

Licenciatura em Engenharia Informática
Unidade Curricular - Desenvolvimento de Aplicações Web
Ano Lectivo: 2022/2023 / 1º semestre / 3º Ano

Relatório
Projecto – Análise/Desenho e **Implementação**

Fábio Gonçalves nº17646

Índice de Capítulos

1	Introdução	5
2	Análise do Sistema.....	5
2.1	Caracterização dos atores	5
2.2	Diagrama de Casos de Uso.....	6
3	Desenho do Sistema.....	9
3.1	Modelação da Base Dados	9
3.1.1	Diagrama E/R.....	9
3.1.2	Modelo Físico	10
3.2	Modelação de Interfaces Gráficas com o Utilizador.....	11
3.2.1	Storyboard(s)	11
3.2.2	Interfaces do caso de uso “Visualizar Catálogo de Jogos”	14
3.2.3	Interfaces do caso de uso “Login/Registo”	14
3.2.4	Interfaces do caso de uso “Visualizar detalhes de um jogo”	15
3.2.5	Interfaces do caso de uso “Finalizar compra do(s) jogo(s)”	15
4	Melhorias efetuadas na análise e desenho do sistema	16
4.1	Layout da homepage (header)	16
4.2	Criação de nova tabela na BD	16
4.3	Adição de colunas nas tabelas na BD.....	16
5	Implementação	17
5.1	Definição da arquitetura do sistema na integração da aplicação com a API	17
5.2	Especificação da interface da API.....	18
5.2.1	Todos os jogos	19
5.2.2	Jogos específico	19

5.2.3	Procurar jogo por nome	20
5.2.4	Procurar jogo através de filtros	20
5.2.5	Jogos em promoção.....	21
5.2.6	Registo	22
5.2.7	Login.....	23
5.2.8	Informação acerca do utilizador.....	24
5.2.9	Adicionar jogo ao carrinho	25
5.2.10	Apagar jogo do carrinho de compras	25
5.2.11	Consultar carrinho de compras	26
5.2.12	Comprar jogos que estão no carrinho de compras	27
5.2.13	Consultar conta	27
5.3	Decisões de implementação	28
5.3.1	Tecnologias usadas	28
5.3.2	Definições de implementação	28
	API.....	28
5.4	Codificação.....	33
4.	Conclusão e Trabalho Futuro.....	36

Índice de Figuras

Figura 1 - Diagrama de Casos de Uso	9
Figura 2 - Diagrama E/R.....	9
Figura 3 – Homepage.....	11
Figura 4 - Interface de login/registo.....	12
Figura 5 - Visualizar detalhes do jogo	12
Figura 6 - Interface de compra (carrinho).....	13
Figura 7 - Wireframe: Homepage	14
Figura 8 - Wireframe: Login/Registo	14
Figura 9 - Wireframe: Visualizar detalhes de determinado jogo selecionado	15
Figura 10 - Wireframe: Carrinho de compras.....	15
Figura 11 - Diagrama de blocos (1)	17
Figura 12 - Diagrama de blocos (2)	17
Figura 13 - Diagrama de blocos (3)	17
Figura 14 - Diagrama de blocos (4)	18
Figura 15 - GET: Todos os jogos.....	19

Figura 16 - Dados da resposta (GET: Todos os jogos)	19
Figura 17 - GET: Jogo específico	19
Figura 18 - Dados da resposta (GET: Jogo específico)	19
Figura 19 - GET: Procurar jogo por nome	20
Figura 20 - Parâmetros (GET: Procurar jogo por nome)	20
Figura 21 - Dados da resposta (GET: Procurar por jogo)	20
Figura 22 - GET: Procurar por jogo através de filtros	20
Figura 23 - Parâmetros (GET: Procurar por jogo através de filtros)	21
Figura 24 - Dados da resposta (GET: Procurar por jogo através de filtros)	21
Figura 25 - GET: Jogos em promoção	21
Figura 26 - Dados da resposta (GET: Jogos em promoção)	21
Figura 27 - POST: Registo.....	22
Figura 28 - Parâmetros (POST: Registo)	22
Figura 29 - Dados da resposta (POST: Registo)	22
Figura 30 - POST: Login	23
Figura 31 - Parâmetros (POST: Login).....	23
Figura 32 - Dados da resposta (POST: Login).....	23
Figura 33 - GET: User info	24
Figura 34 - Inserir token	24
Figura 35 - Dados da resposta (GET: User info)	24
Figura 36 - POST: Adicionar jogo ao carrinho.....	25
Figura 37 - Parâmetros (POST: Adicionar jogo ao carinho)	25
Figura 38 - Dados da resposta (POST: Adicionar jogo ao carrinho)	25
Figura 39 - DELETE: Apagar jogo no carrinho de compras	25
Figura 40 - Dados de resposta (DELETE: Apagar jogo no carrinho de compras)	26
Figura 41 - GET: Consultar carrinho de compras.....	26
Figura 42 - Dados da resposta (GET: Consultar carrinho de compras)	26
Figura 43 - POST: Comprar jogos	27
Figura 44 - Dados da resposta (POST: Comprar jogos)	27
Figura 45 - GET: Consultar conta.....	27
Figura 46 - Dados da resposta (POST: Consultar conta).....	27
Figura 47 – Cookies	28
Figura 48 - Campos de palavra-passe	29
Figura 49 - Campo da palavra-passe na base de dados.....	29
Figura 50 - Homepage	29
Figura 51 - Procurar jogo por nome.....	30

Figura 52 - Procurar por jogos através de filtros	30
Figura 53 - Jogos em promoção.....	31
Figura 54 - Consultar detalhes de um jogo em específico	31
Figura 55 - Login/Registo.....	32
Figura 56 - Carrinho de compras	32
Figura 57 - Conta do utilizador.....	33
Figura 58 - AuthController: signin	34
Figura 59 - AuthController: signup.....	34
Figura 60 - game.blade.php	35

Índice de Tabelas

Tabela 1 - Tabela de especificação de requisitos (Visualizar Jogos).....	7
Tabela 2 - Tabela de especificação de requisitos (Comprar Jogo(s))	8
Tabela 3 - Tabela "Users" (BD).....	10
Tabela 4 - Tabela "Cart" (BD)	10
Tabela 5 - Tabela "Games" (BD).....	10
Tabela 6 - Tabela "Orders" (BD)	10
Tabela 7 – Endpoints	18

1 Introdução

Este projeto no âmbito da unidade curricular de desenvolvimento de aplicações web tem como objetivo desenvolver e demonstrar as competências dos alunos no desenvolvimento de aplicações web do lado do servidor, para tal, como tema de projeto foi escolhido realizar a fase de análise/desenho e a implementação de uma loja online de videojogos, em que neste caso a área de aplicação a desenvolver será o comércio eletrónico.

Este projeto tem como objetivo permitir aos clientes que visitem o site, visualizar os jogos disponíveis no catálogo do website e se assim o desejaram comprar algum dos jogos e também do lado do administrador o mesmo poder gerir o catálogo de jogos em que poderá efetuar tarefas como, adicionar, remover ou até mesmo editar determinados jogos.

As tarefas que serão implementadas neste projeto são:

- **Tarefa 1** – Inserir jogo;
- **Tarefa 2** – Modificar jogo;
- **Tarefa 3** – Consultar catálogo de jogos/jogo;
- **Tarefa 4** – Comprar jogo.

No caso deste projeto serão descritos os procedimentos para a implementação da tarefa 3 e tarefa 4 que foram as que foram implementadas através da *framework* [Laravel](#).

As outras duas pelo meu colega de trabalho que realizou a implementação em ASP.NET.

2 Análise do Sistema

Em seguida é descrito o levantamento e a especificação de requisitos e de características de elementos do sistema que foram desenvolvidos durante esta fase de análise do sistema.

2.1 Caracterização dos atores

Este sistema tem como utilizadores alvo qualquer utilizador cujo seu interesse seja comprar um videojogo online, ou apenas consultar que jogos estão à venda atualmente no nosso catálogo.

O sistema a desenvolver tem maioritariamente objetivo de captar o interesse de clientes cujo as suas faixas etárias sejam jovens visto que o interesse em videojogos geralmente seja maior, mas tendo em conta que a faixa etária de adultos também seja importante.

O público-alvo deste sistema será também um utilizador que tenha alguma experiência a utilizar websites de compras online, apesar da interface deste sistema ser simples em termos de design e fácil utilização pode tornar-se um pouco confusa para utilizadores novos que não estejam muito familiarizados com a compra de videojogos online.

Caracterização dos atores:

- Cliente – Cenário de utilização da Tarefa 3 (Visualizar jogos disponíveis)

Tiago tem 25 anos e trabalha na área de logística de um armazém, nos seus tempos livres é um jogador de videojogos avido.

Após acabar o seu dia de trabalho, chega a casa e sente uma vontade de jogar um novo jogo, então, Tiago consulta o seu website favorito de vendas de videojogos, e começa a visualizar os jogos disponíveis e os seus preços abrindo a página específica de cada um para ler também a descrição de maneira a ajudar-lhe a fazer a escolha.

- Cliente – Cenário de utilização da Tarefa 4 (Comprar um jogo)

Após Tiago ter pensado um pouco sobre qual jogo comprar, dirige-se à página específica do mesmo, e carrega em comprar, sendo redirecionado à página de pagamento, insere os seus dados, e por fim quando o pagamento é processado, é apresentada a chave do jogo.

Tiago vai até à plataforma escolhida e ativa a sua chave, podendo assim jogar o seu jogo.

2.2 Diagrama de Casos de Uso

Tabelas de especificação de requisitos:

Nome do Caso de Uso	Visualizar Jogos
Descrição Resumida	O caso de uso tem como objetivo todos os utilizadores do sistema conseguirem visualizar os jogos disponíveis no catálogo do website.
Fluxo de Eventos	O utilizador visualiza as diferentes plataformas de jogos, os diferentes géneros de cada um deles e após as determinadas filtragens o utilizador visualiza os jogos.

Fluxo Básico	O utilizador consegue visualizar e utilizar o sistema.
Fluxos Alternativos	O utilizador ao visualizar informação acerca de um determinado jogo e este ter informação alterada ou até mesmo excluída nesse determinado momento.
Requisitos Especiais	<ul style="list-style-type: none"> • Requisito de usabilidade: Interface do sistema simples e fácil de interpretar e navegar.
Pré-Condições	Nenhuma pré-condição necessária para este caso de uso.
Pós-Condições	<ul style="list-style-type: none"> • Navegar pelo catálogo de jogos (de cima para baixo); <ul style="list-style-type: none"> o Selecionar a plataforma; o Selecionar o género; o Ordenar por preço. • Consultar mais detalhes acerca de um determinado jogo para uma informação mais detalhada.

Tabela 1 - Tabela de especificação de requisitos (Visualizar Jogos)

Nome de Caso de Uso	Comprar Jogo(s)
Descrição Resumida	O caso de uso tem como objetivo todos os utilizadores do sistema conseguirem comprar um determinado jogo.

Fluxo de Eventos	O utilizador seleciona o jogo pretendido, carrega no botão de comprar, introduz o método de pagamento, clica em finalizar, após tudo bem sucedido o utilizador recebe a chave para ativar o jogo em determinada plataforma.
Fluxo Básico	O utilizador consegue efetuar uma compra com sucesso.
Fluxos Alternativos	<p>O utilizador ao tentar finalizar a compra do jogo não é possível realizar essa ação. Ex:</p> <ul style="list-style-type: none"> • Método de pagamento recusado; • Último jogo disponível ter sido comprado no momento de finalizar a compra.
Requisitos Especiais	<ul style="list-style-type: none"> • Requisito de usabilidade: Interface do sistema simples e fácil de interpretar e navegar. Diversos métodos de pagamento, para dar uma maior escolha e facilidade ao utilizador.
Pré-Condições	Utilizador tenha o login efetuado no sistema.
Pós-Condições	<ul style="list-style-type: none"> • Selecionar o jogo que se pretende comprar; • Introduzir os detalhes do método de pagamento; • Consultar toda a informação acerca do jogo comprado através da conta; • Terminar sessão no sistema.

Tabela 2 - Tabela de especificação de requisitos (Comprar Jogo(s))

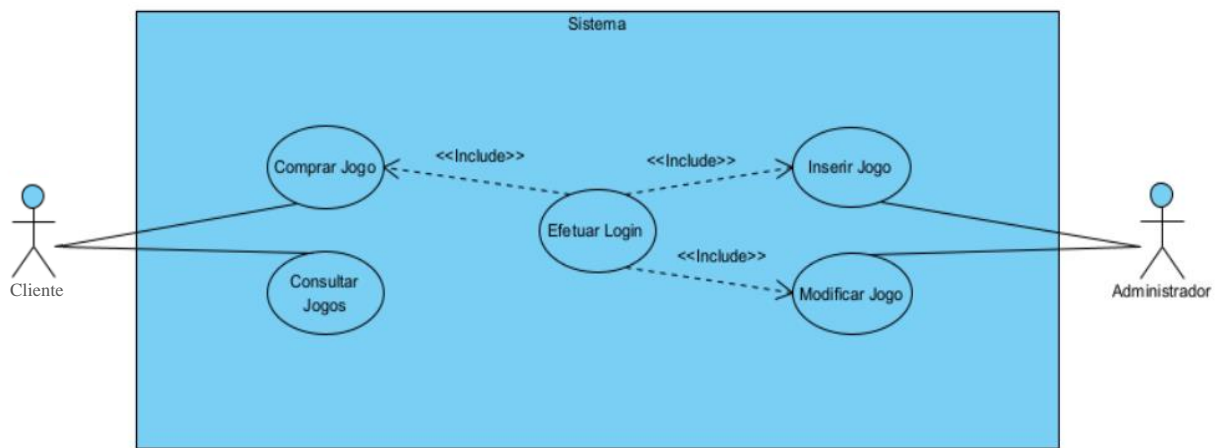


Figura 1 - Diagrama de Casos de Uso

3 Desenho do Sistema

Em seguida são descritos os modelos propostos de interfaces gráficas com o utilizador e da base de dados.

3.1 Modelação da Base Dados

Em seguida são descritos os modelos E/R e físico da base de dados contendo todas as entidades de dados/tabelas que fazem parte de todos os requisitos do sistema, independentemente de alguns desses dados não serem geridos através das interfaces gráficas com o utilizador que fazem parte do protótipo deste projeto.

3.1.1 Diagrama E/R

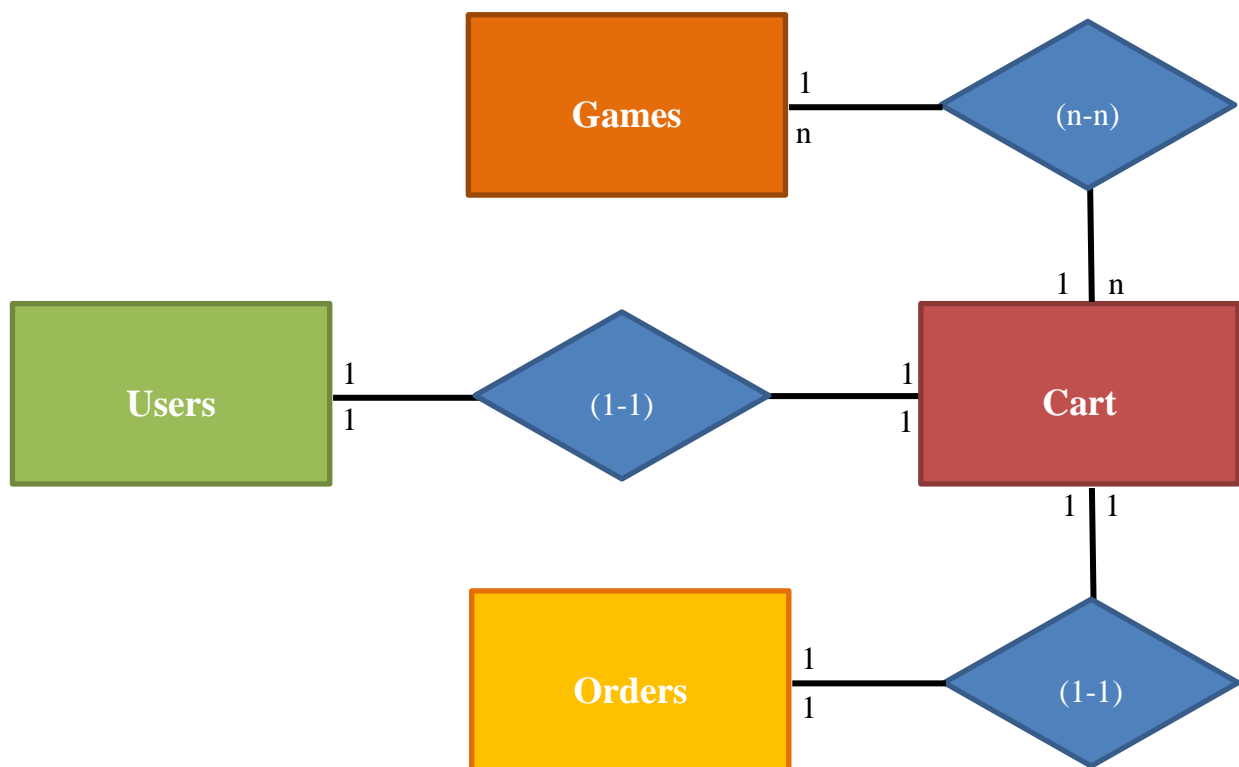
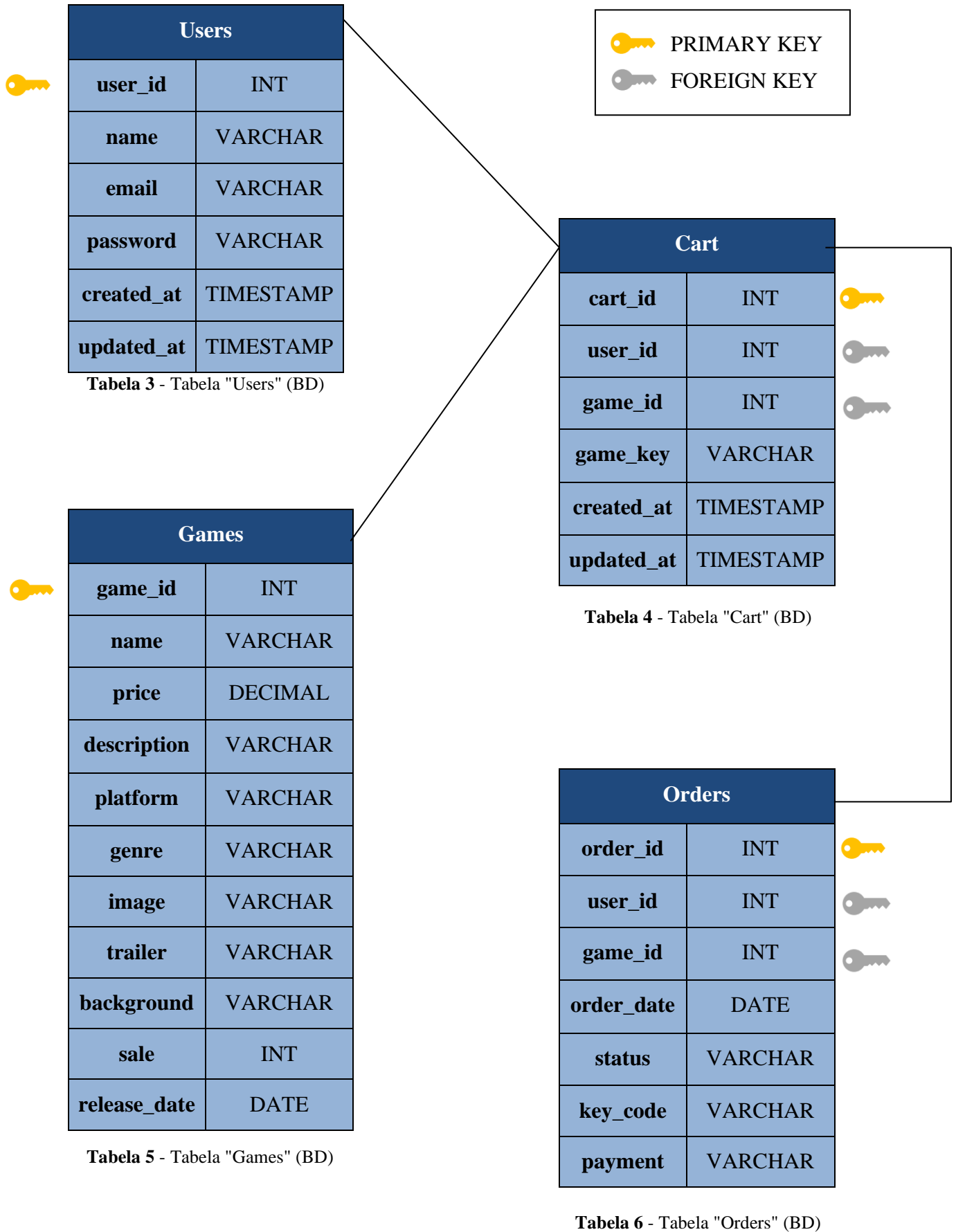


Figura 2 - Diagrama E/R

3.1.2 Modelo Físico



3.2 Modelação de Interfaces Gráficas com o Utilizador

Em seguida são descritos os modelos desenvolvidos relativos ao protótipo de baixa/média fidelidade das interfaces gráficas com o utilizador, compostos por *storyboards* e pela proposta de ecrãs (*wireframes*) da aplicação. Os *storyboards* desenvolvidos permitem perceber genericamente a sequência de ações de navegação do utilizador no sistema. As propostas de interfaces gráficas com o utilizador são agrupadas por caso de uso e mostram o *layout* e o conteúdo de apresentação de cada página Web, bem como, a forma como o utilizador interage com ela.

3.2.1 Storyboard(s)

Visualizar catálogo de jogos

Está é a página inicial, onde o utilizador consegue já visualizar os jogos disponíveis (com ou sem login iniciada, apenas terá de iniciar sessão com as suas credenciais caso pretenda comprar um ou mais jogos).

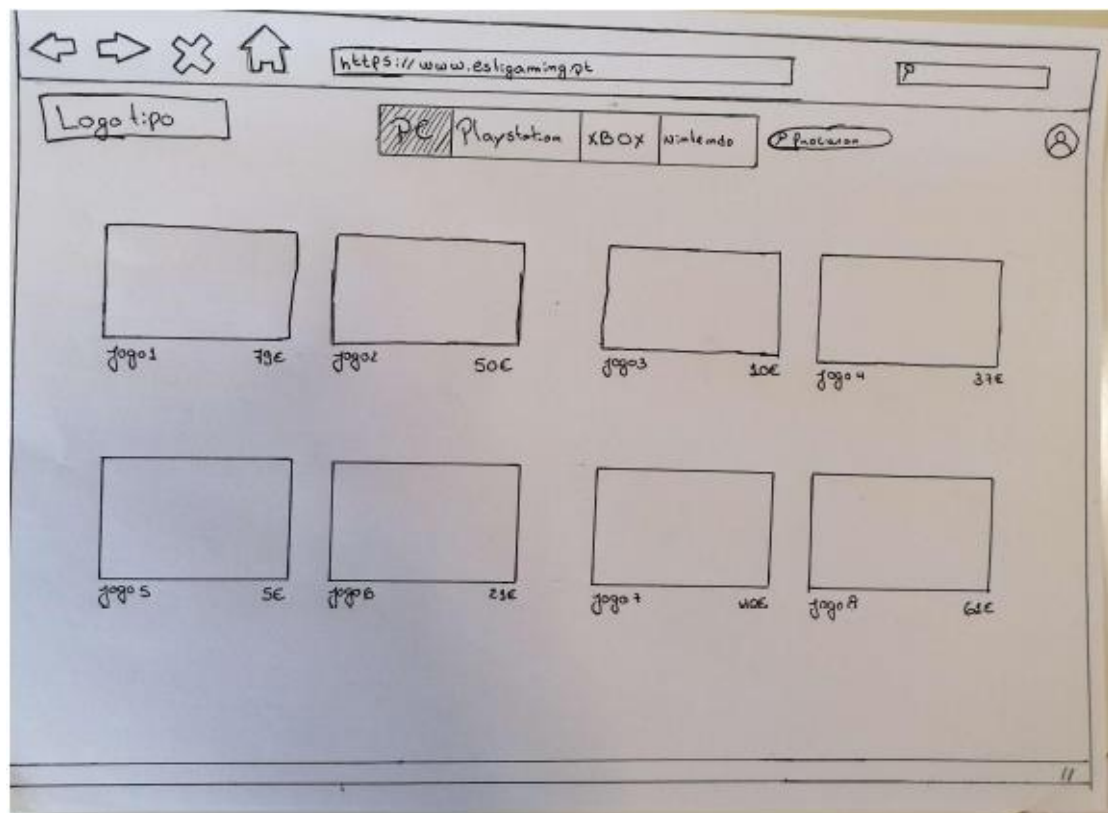


Figura 3 – Homepage

Iniciar Sessão

Ao carregar no ícone da conta, o utilizador terá acesso à interface onde pode iniciar sessão caso já tenha conta, introduzindo o seu email e palavra-passe. Ou se ainda não tiver conta pode também aqui realizar o registo.

The image shows a hand-drawn interface for login and registration, divided into two columns by a vertical line.

Left Column (Login):

- Title: "Iniciar Sessão"
- Form fields: "Email" (with a checkmark icon) and "Password" (with a key icon).
- Text: "Esqueceu a PW" (Forgot the password).
- Button: "Entrar" (Login).

Right Column (Registration):

- Title: "Registar"
- Form fields: "User name" (with a person icon), "Email" (with a checkmark icon), and "Password".
- Button: "Registar" (Register).

Figura 4 - Interface de login/registo

Informação detalhada do jogo carregado

The image shows a hand-drawn interface for viewing game details, resembling a web browser window.

Browser Header:

- Navigation icons: back, forward, close, and home.
- Address bar: "https://www.estigaming.pt"
- Search bar: empty.

Game Information Section:

- Logo: "LOGOTIPO"
- Platform tabs: "PC" (selected), "Playstation", "XBOX", "Nintendo", "Provision".
- Image placeholder: "Imagem" (with a vertical line).
- Game details: "Nome do jogo", "Plataforma: steam", "Gêneros: Ação", "79€", and a "Comprar" button.

Description Section:

- Title: "Sobre o jogo:"
- Text: "Informações sobre o jogo." followed by several horizontal lines for text.

Figura 5 - Visualizar detalhes do jogo

Carrinho com os jogos selecionados

Por fim existe a interface final de compra, que será apresentado logo após o utilizador carregar no comprar em algum jogo, continuará presente a imagem do(s) jogo(s) e algumas informações de maneira que o utilizador tenha a certeza que não está a comprar o jogo errado, e terá um dropdown com os vários métodos de pagamento possíveis, após selecionar o pretendido, pode carregar no comprar e será redirecionado para o método de pagamento escolhido

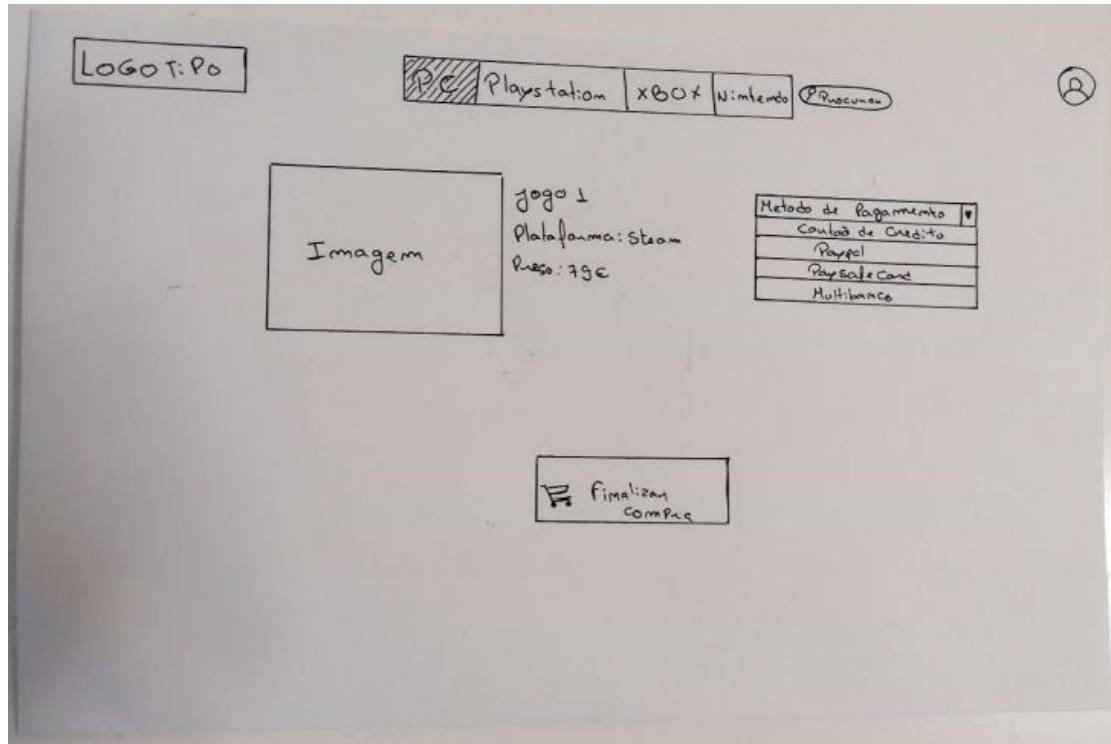


Figura 6 - Interface de compra (carrinho)

3.2.2 Interfaces do caso de uso “Visualizar Catálogo de Jogos”

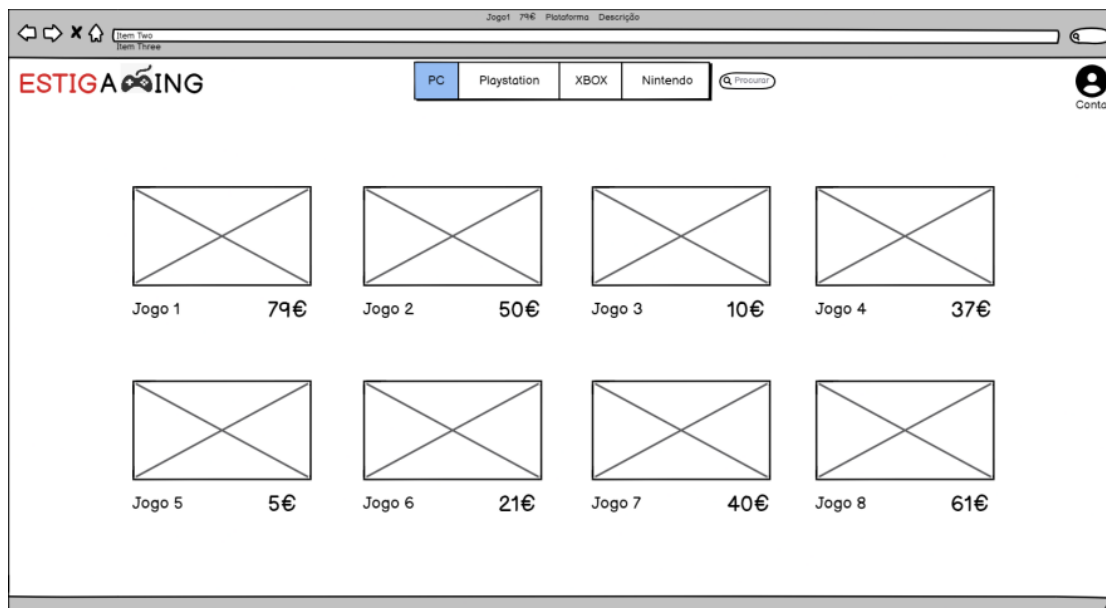


Figura 7 - Wireframe: Homepage

3.2.3 Interfaces do caso de uso “Login/Registo”



Figura 8 - Wireframe: Login/Registo

3.2.4 Interfaces do caso de uso “Visualizar detalhes de um jogo”

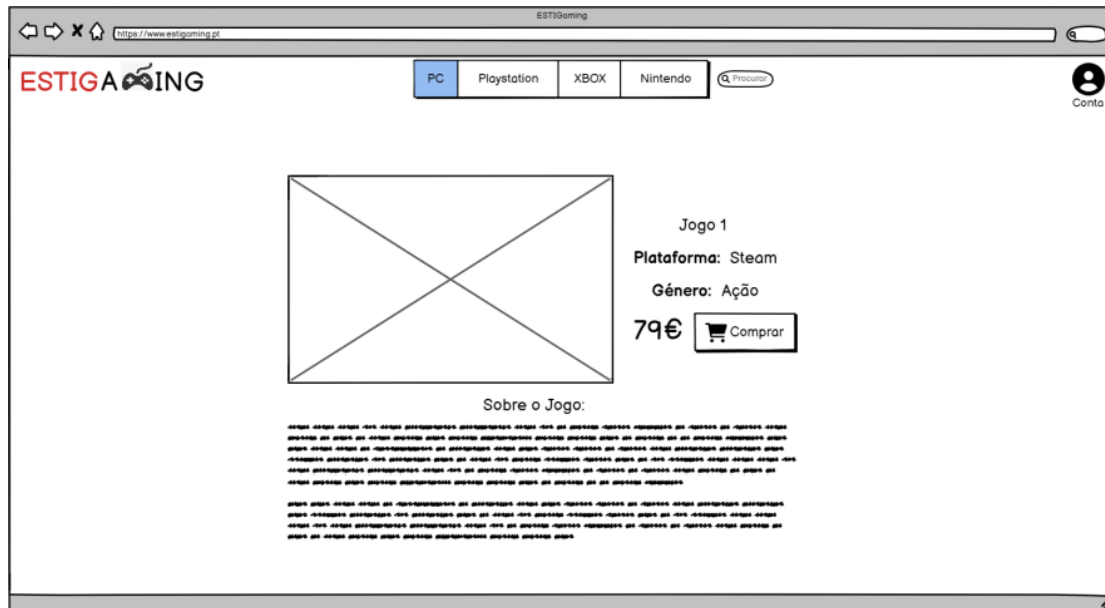


Figura 9 - Wireframe: Visualizar detalhes de determinado jogo selecionado

3.2.5 Interfaces do caso de uso “Finalizar compra do(s) jogo(s)”

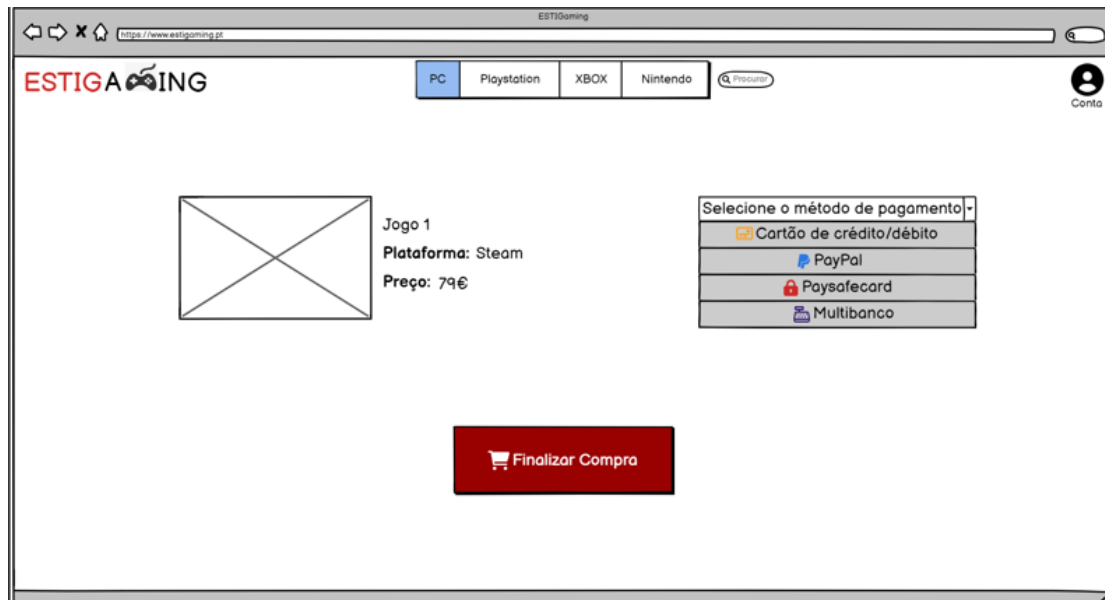


Figura 10 - Wireframe: Carrinho de compras

4 Melhorias efetuadas na análise e desenho do sistema

Neste capítulo são descritas as principais melhorias efetuadas após a defesa desta fase do projeto.

4.1 Layout da homepage (header)

Após criação do design das views (.blade.php) através dos desenhos dos storyboards e wireframes na fase de análise, após a sua implementação no projeto foi alterado o seu layout maioritariamente do *header* visto que após mostrar as interfaces a amigos para darem a sua opinião e tentar utilizar o website para conseguir obter melhorias e o seu feedback, muitos deles sugeriram que como existem poucas opções de escolha nas interfaces deveria juntar/centrar as opções disponíveis e assim tornar-se mais intuitivo ao utilizador do que fazer.

Então decidi centrar as opções disponíveis no website que se encontram no *header*, como por exemplo (*input* de procura por nome de jogo, filtros para procurar um jogo através da sua categoria, promoções, carrinho e botão de login/conta do utilizador).

4.2 Criação de nova tabela na BD

Após a apresentação do projeto na fase de análise/desenho foi proposto pelo docente da unidade curricular a criação da tabela do carrinho “cart” caso o utilizador queira adicionar mais do que um jogo, visto que no esquema apresentado nessa fase existiam apenas três tabelas (“users”, “games” e “orders”) que tinham como objetivo o jogo selecionar um jogo e ser “forçado” a comprar esse jogo, e assim com a sugestão do docente permite ao utilizador adicionar vários jogos, estes que irão ficar armazenados nesta tabela e depois quando o utilizador tiver adicionado todos os jogos que pretende poderá então aceder ao carrinho com os jogos que foram adicionados e por fim finalizar a compra.

4.3 Adição de colunas nas tabelas na BD

Foram adicionados mais campos a cada uma das tabelas existentes na base de dados desde a fase de análise/desenho até esta fase de implementação, esses campos foram adicionados para resolver alguns dos problemas que estavam a originar na criação de *queries* na parte do *model* e para facilitar a implementação do sistema e o seu melhor funcionamento.

5 Implementação

Neste capítulo são descritas as principais decisões e ações desenvolvidas na parte da implementação técnica deste projeto.

5.1 Definição da arquitetura do sistema na integração da aplicação com a API

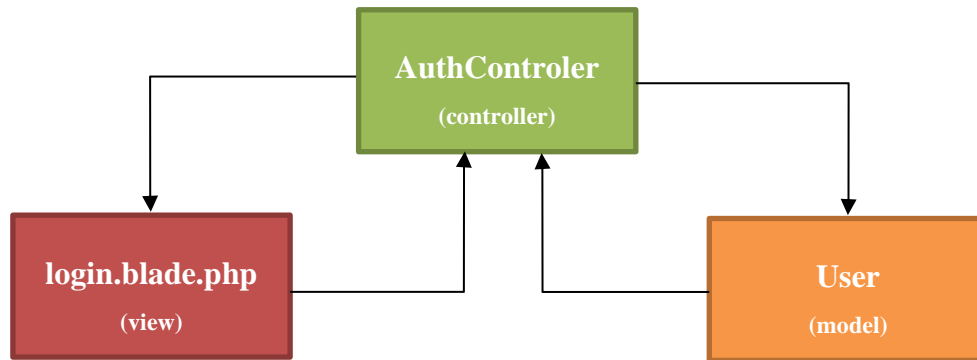


Figura 11 - Diagrama de blocos (1)

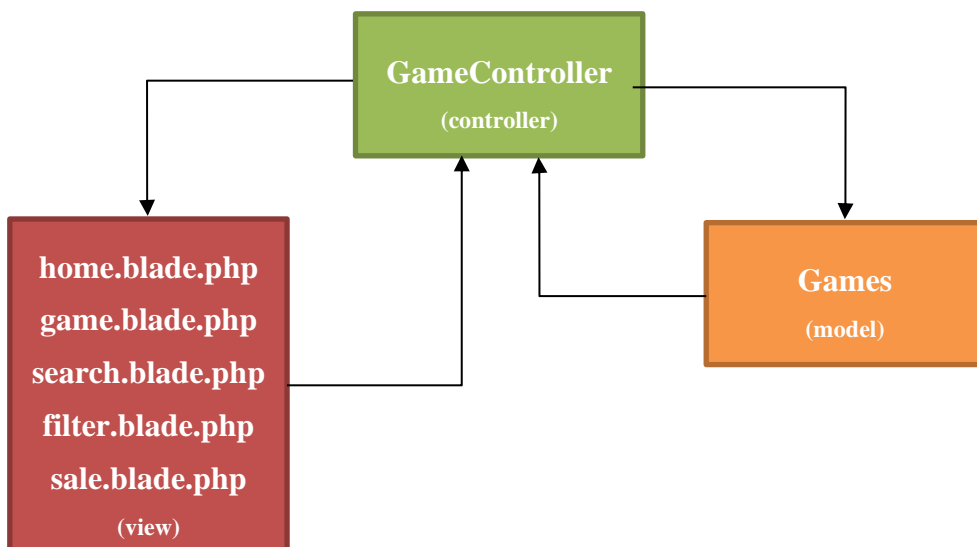


Figura 12 - Diagrama de blocos (2)

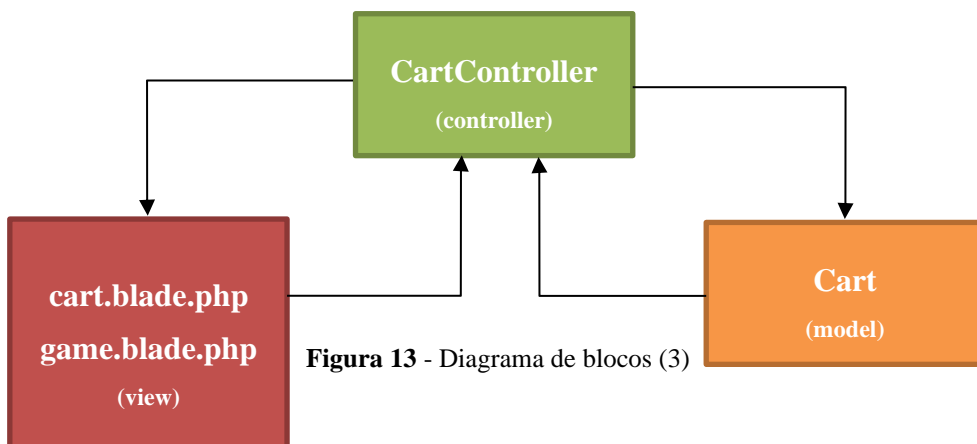


Figura 13 - Diagrama de blocos (3)

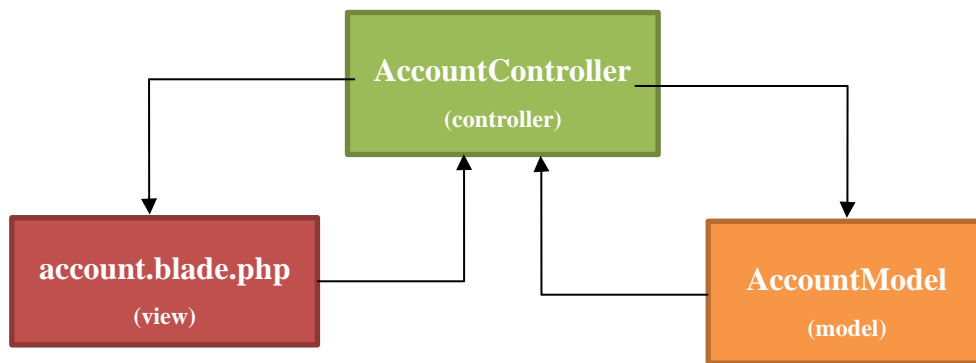


Figura 14 - Diagrama de blocos (4)

5.2 Especificação da interface da API

Para especificar a estrutura da API REST desenvolvida será usado o [Postman](#) que foi utilizado inicialmente na implementação dos seus endpoints e em que foram testados os mesmos.

GET	http://localhost:8000/api/games	Todos os jogos
GET	http://localhost:8000/api/game{id}	Jogo específico
GET	http://localhost:8000/api/search	Procurar jogo por nome
GET	http://localhost:8000/api/filter	Procurar jogo por filtros
GET	http://localhost:8000/api/sale	Jogos em promoção
GET	http://localhost:8000/api/cart	Carrinho de compras
GET	http://localhost:8000/api/user	Informações do utilizador
GET	http://localhost:8000/api/account	Conta do utilizador
POST	http://localhost:8000/api/login	Login
POST	http://localhost:8000/api/register	Registo
POST	http://localhost:8000/api/addToCart	Adicionar jogo ao carrinho
POST	http://localhost:8000/api/cart/buy	Comprar jogos do carrinho
DELETE	http://localhost:8000/api/cart/deleteFromCart/{id}	Apagar jogo do carrinho

Tabela 7 – Endpoints

5.2.1 Todos os jogos

Permite obter a lista de todos os jogos que existem no sistema.

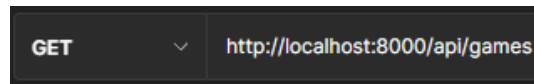


Figura 15 - GET: Todos os jogos

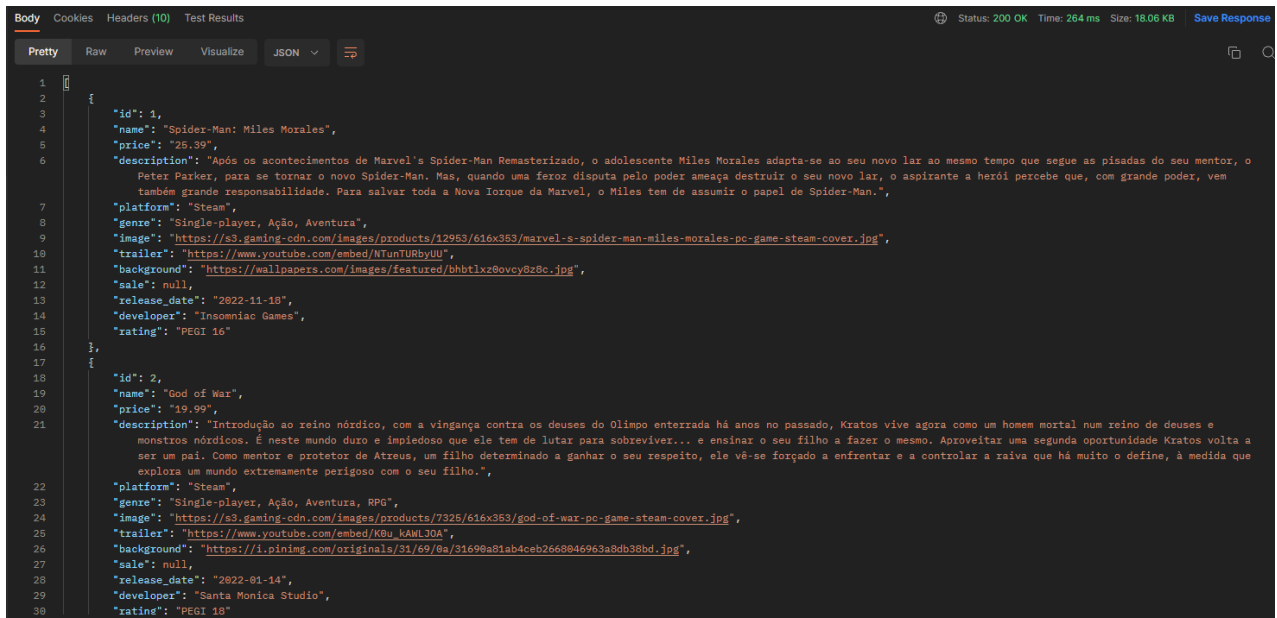


Figura 16 - Dados da resposta (GET: Todos os jogos)

5.2.2 Jogos específico

Permite obter um determinado jogo através do seu id, devolver todos os seus detalhes apenas do jogo correspondente a esse id.

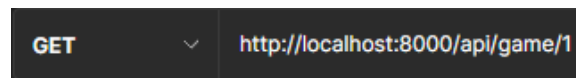


Figura 17 - GET: Jogo específico

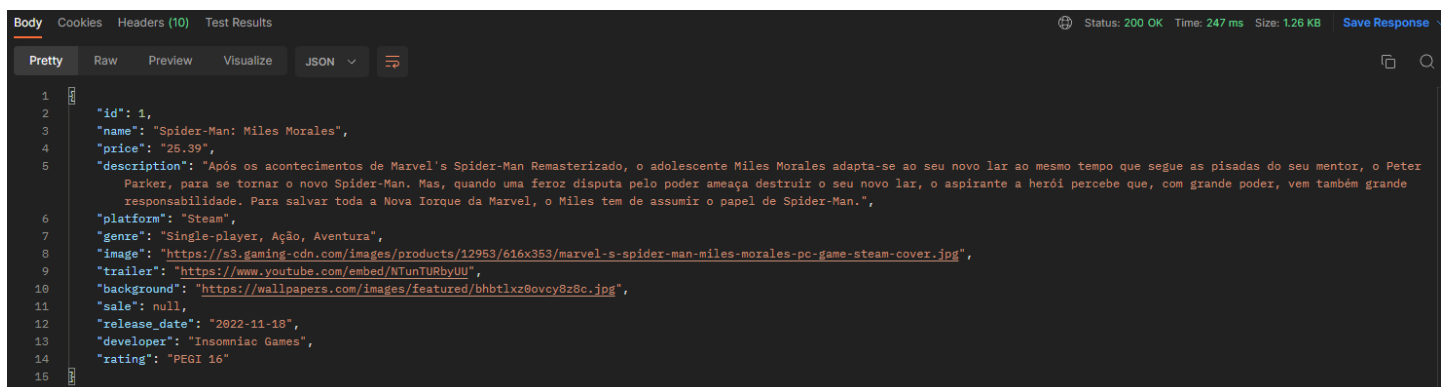


Figura 18 - Dados da resposta (GET: Jogo específico)

5.2.3 Procurar jogo por nome

Permite obter um determinado jogo através do nome colocado como parâmetro e como resposta serão devolvidos jogos que contenham esse nome/palavras no seu título.

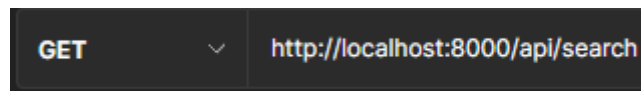


Figura 19 - GET: Procurar jogo por nome

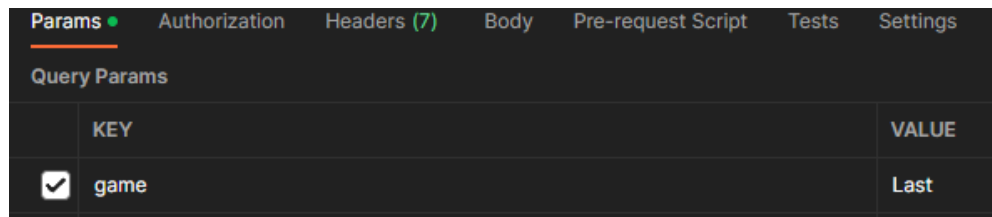


Figura 20 - Parâmetros (GET: Procurar jogo por nome)



Figura 21 - Dados da resposta (GET: Procurar por jogo)

5.2.4 Procurar jogo através de filtros

É possível encontrar jogos através de diferentes filtros:

- Plataforma (Steam, Playstation e Xbox);
- Género (Single-player, Ação, Aventura, RPG e Online);
- Ordenar por preço.

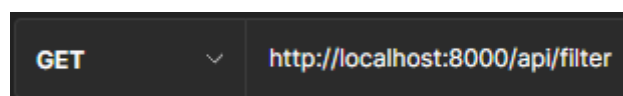


Figura 22 - GET: Procurar por jogo através de filtros

Params • Authorization Headers (6) Body Pre-request Script Tests Settings		
Query Params		
	KEY	VALUE
<input checked="" type="checkbox"/>	platform	Steam
<input checked="" type="checkbox"/>	genre	Ação
<input checked="" type="checkbox"/>	orderby	ASC

Figura 23 - Parâmetros (GET: Procurar por jogo através de filtros)

Body Cookies Headers (10) Test Results		Status: 200 OK Time: 297 ms Size: 6.82 KB Save Response
Pretty Raw Preview Visualize JSON		
47	{	
48	"id": 13,	
49	"name": "Dead Space",	
50	"price": "68",	
51	"description": "O clássico de sobrevivência de terror e ficção científica Dead Space regressa, totalmente reconstruído do zero para oferecer uma experiência mais profunda e imersiva. Este remake traz uma fidelidade visual assombrosa, áudio atmosférico cheio de suspense e melhorias na jogabilidade, mantendo-se fiel à visão emocionante do jogo original. Isaac Clarke é um engenheiro comum numa missão para reparar uma vasta nave de mineração, a USG Ishimura, apenas para descobrir que algo correu terrivelmente mal. A tripulação da nave foi massacrada e a amada parceira de Isaac, Nicole, está perdida em algum lugar a bordo.",	
52	"platform": "Steam",	
53	"genre": "Single-player, Ação, Aventura",	
54	"image": "https://s2.gaming-cdn.com/images/products/9894/616x353/dead-space-pc-jogo-origin-cover.jpg",	
55	"trailer": "https://www.youtube.com/embed/cTDJNZ9cK1w",	
56	"background": "https://cdn.wallpapersafari.com/73/2/04AJPz.jpg",	
57	"sale": null,	
58	"release_date": "2023-01-27",	
59	"developer": "MOTIVE",	
60	"rating": "PEGI 18"	
61	}	

Figura 24 - Dados da resposta (GET: Procurar por jogo através de filtros)

5.2.5 Jogos em promoção

É disponibilizada a lista de todos os jogos que se encontram em promoção.

GET
▼
http://localhost:8000/api/sale

Figura 25 - GET: Jogos em promoção

Body Cookies Headers (10) Test Results		Status: 200 OK Time: 269 ms Size: 5.9 KB Save Response
Pretty Raw Preview Visualize JSON		
1	{	
2	"id": 3,	
3	"name": "Dead by Daylight",	
4	"price": "29",	
5	"description": "A morte não é uma saída. Dead by Daylight é um jogo de horror de multijogadores (4x1). Um jogador assume o papel do Assassino enquanto os outros quatro jogadores jogam como Sobreviventes e tentam fugir do Assassino - ou serão pegos, torturados e assassinados. Os Sobreviventes jogam em terceira pessoa e têm como vantagem uma melhor percepção situacional. O Assassino joga em primeira pessoa e está mais focado na presa. A cada encontro, o objetivo dos Sobreviventes é fugir do Território de Abate sem ser pego pelo Assassino. Isso é mais difícil do que parece - ainda mais em um cenário que muda a cada partida.",	
6	"platform": "Steam",	
7	"genre": "Online PVP, Ação",	
8	"image": "https://s3.gaming-cdn.com/images/products/1984/616x353/dead-by-daylight-pc-game-steam-cover.jpg",	
9	"trailer": "https://www.youtube.com/embed/3GhIXL03u18",	
10	"background": "https://images5.alphacoders.com/778/778923.jpg",	
11	"sale": "78",	
12	"release_date": "2016-06-14",	
13	"developer": "Behaviour Interactive Inc.",	
14	"rating": "PEGI 18"	
15	}	

Figura 26 - Dados da resposta (GET: Jogos em promoção)

5.2.6 Registo

Permite criar uma conta no sistema através do preenchimento dos campos:

- Nome;
- Email;
- Palavra-passe;
- Confirmação de palavra-passe.

Após o registo uma token será gerada para esse utilizador em que poderá usar para se autenticar em endpoints cujo esses sejam precisos autenticação obrigatório para serem acedidos.

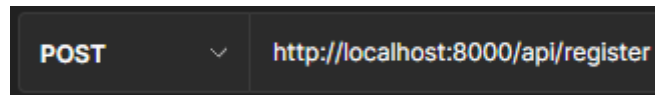
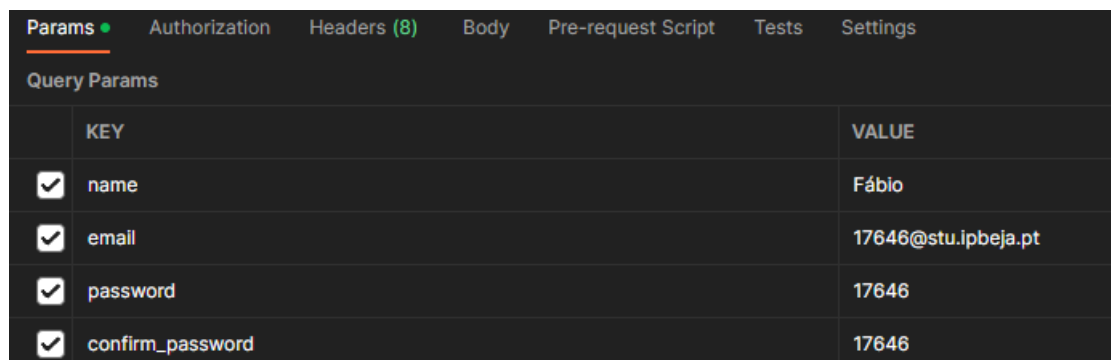


Figura 27 - POST: Registo

A screenshot of a REST client interface showing the 'Query Params' tab. It displays a table with four rows, each representing a query parameter. The first column is 'KEY' and the second is 'VALUE'. Each row has a checkmark in the first column. The parameters are: name (Fábio), email (17646@stu.ipbeja.pt), password (17646), and confirm_password (17646).

	KEY	VALUE
<input checked="" type="checkbox"/>	name	Fábio
<input checked="" type="checkbox"/>	email	17646@stu.ipbeja.pt
<input checked="" type="checkbox"/>	password	17646
<input checked="" type="checkbox"/>	confirm_password	17646

Figura 28 - Parâmetros (POST: Registo)

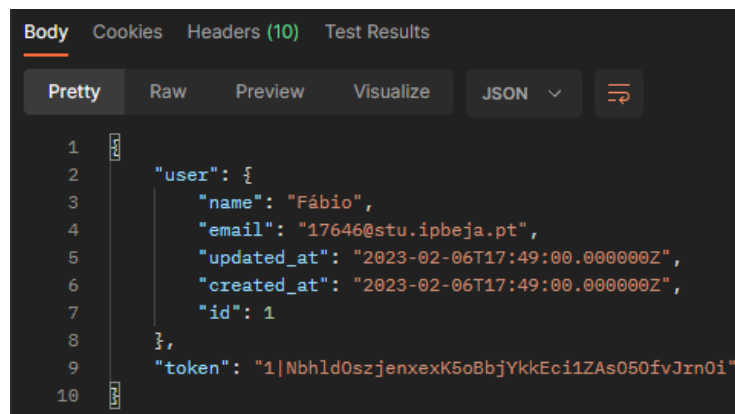


Figura 29 - Dados da resposta (POST: Registo)

5.2.7 Login

Efetuar o início de sessão para conseguir utilizar todas as funcionalidades do sistema, principalmente comprar um jogo.

Cada vez que o utilizador efetua login no sistema é gerada uma nova *token* associada ao seu início de sessão que pertence exclusivamente a ele mesmo e irá permitir que se autentique em outras funcionalidades do sistema.

A *token* só é removida dessa sessão após o utilizador efetuar o *logout* do sistema.

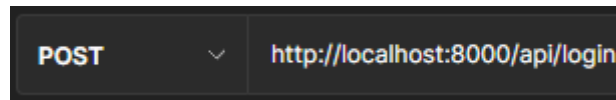


Figura 30 - POST: Login

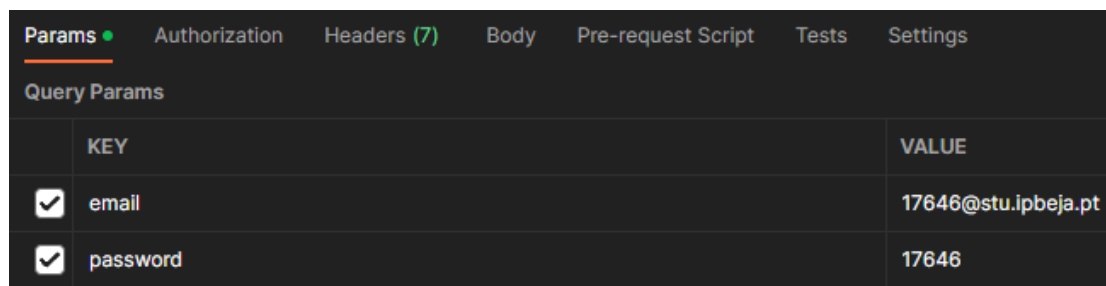


Figura 31 - Parâmetros (POST: Login)

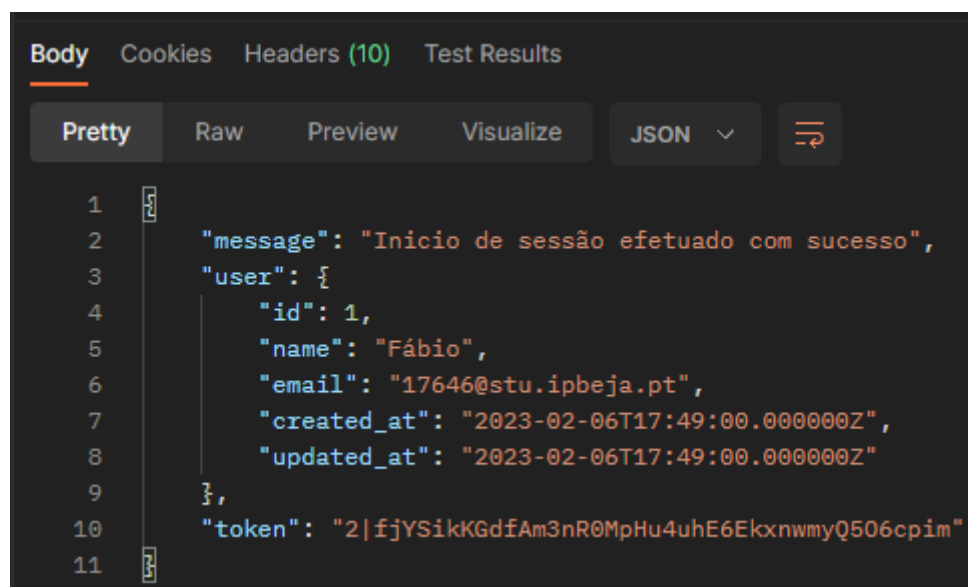


Figura 32 - Dados da resposta (POST: Login)

5.2.8 Informação acerca do utilizador

Possibilidade de obter mais informações acerca dos dados mais detalhados do utilizador que estiver com a login efetuada, é possível aceder a essa informação através da sua *token*.

Caso não seja disponibilizada nenhuma *token*, a resposta devolvida pelo *endpoint* será de “*Unauthenticated*”.

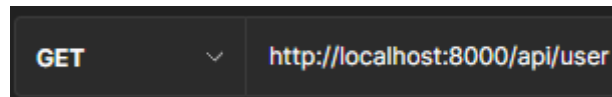


Figura 33 - GET: User info

Daqui em diante todas as funcionalidades que requerem o utilizador estar autenticado será necessário inserir sempre a sua *token* de utilizador para que as consiga aceder, caso contrário irá ser devolvida a resposta de “*Unauthenticated*” e não será possível aceder a essa funcionalidade.

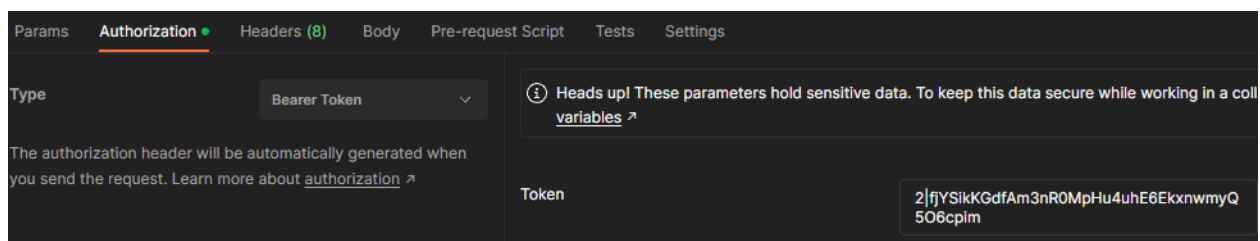


Figura 34 - Inserir token

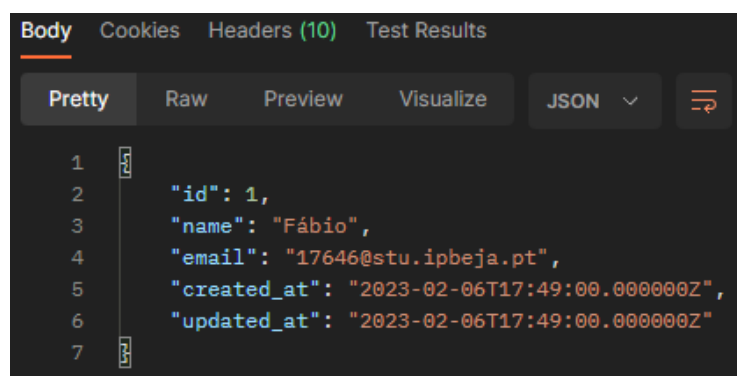


Figura 35 - Dados da resposta (GET: User info)

5.2.9 Adicionar jogo ao carrinho

Permite adicionar um jogo ao carrinho através do seu “*id*” que pode ser obtido ao visualizar através de um dos *GET*’s mostrados previamente que permitem visualizar os jogos associados aos seus ids.

O utilizador precisa de disponibilizar a sua *token* para aceder a esta funcionalidade, caso não seja disponibilizada nenhuma *token*, a resposta devolvida pelo *endpoint* será de “*Unauthenticated*”.

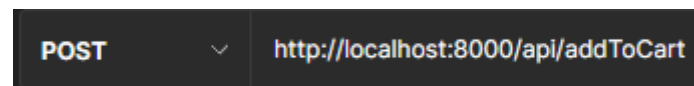


Figura 36 - POST: Adicionar jogo ao carrinho

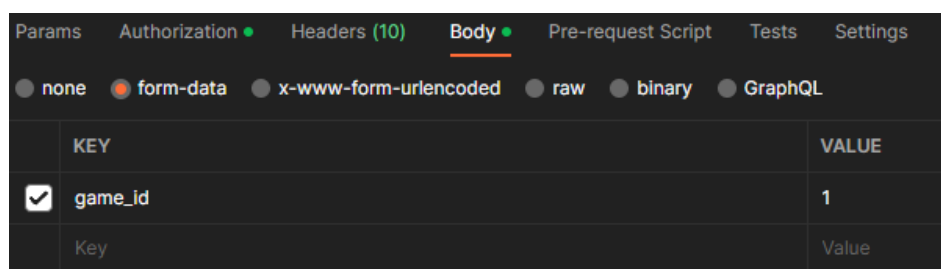


Figura 37 - Parâmetros (POST: Adicionar jogo ao carrinho)

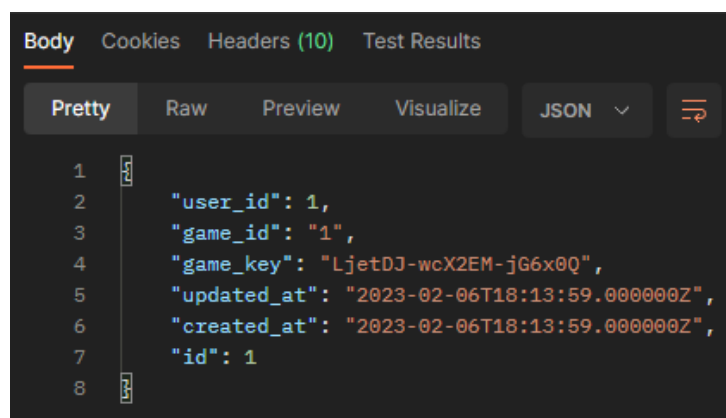


Figura 38 - Dados da resposta (POST: Adicionar jogo ao carrinho)

5.2.10 Apagar jogo do carrinho de compras

Permite ao utilizador apagar um jogo existente no seu carrinho de compras através do “*id*”, para tal o utilizador precisa de estar autenticado através da sua *token*.

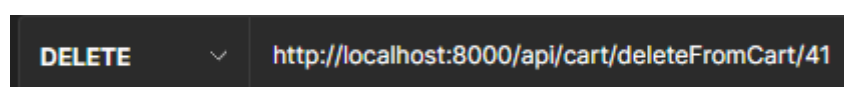


Figura 39 - DELETE: Apagar jogo no carrinho de compras

Possibilidades de *output*:

- **1** – Um jogo foi apagado com sucesso do carrinho;
- **0** – Nenhum jogo foi apagado (*id* incorreta).

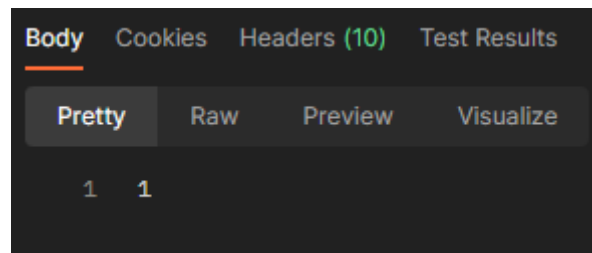


Figura 40 - Dados de resposta (DELETE: Apagar jogo no carrinho de compras)

5.2.11 Consultar carrinho de compras

Permite consultar o carrinho de comprar e verificar os jogos que estão nele, através da sua *token* de autenticação.

O utilizador precisa de disponibilizar a sua *token* para aceder a esta funcionalidade, caso não seja disponibilizada nenhuma *token*, a resposta devolvida pelo *endpoint* será de “*Unauthenticated*”.

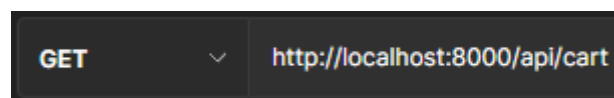


Figura 41 - GET: Consultar carrinho de compras

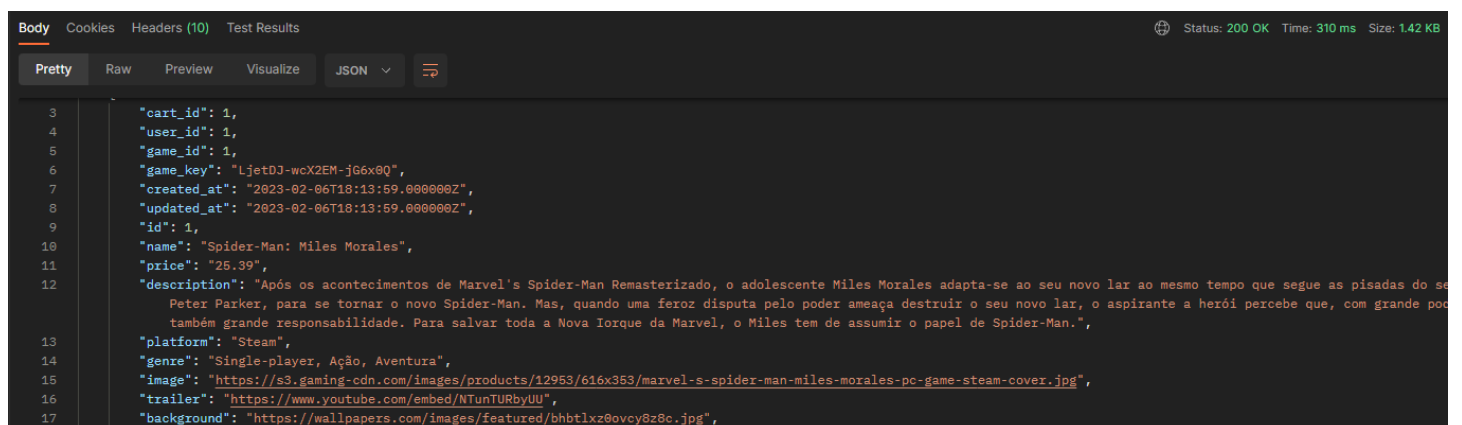


Figura 42 - Dados da resposta (GET: Consultar carrinho de compras)

5.2.12 Comprar jogos que estão no carrinho de compras

Permite comprar todos os jogos que se encontram no carrinho de compras, desde que o utilizador esteja autenticado com a sua *token* pessoal.

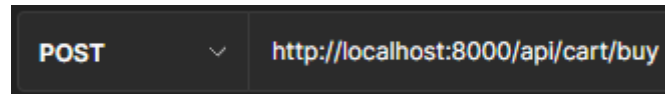


Figura 43 - POST: Comprar jogos

Introdução do parâmetro do método de pagamento.

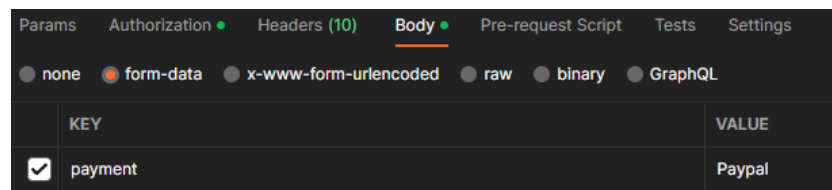


Figura 44 - Dados da resposta (POST: Comprar jogos)

5.2.13 Consultar conta

Permite consultar informações sobre a conta do utilizador com sessão iniciada (*token*), e ver o seu histórico de jogos comprados, assim como mais detalhes sobre o mesmo.

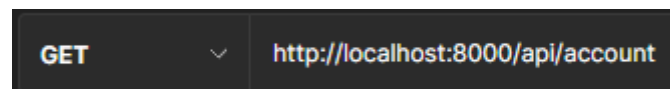


Figura 45 - GET: Consultar conta



Figura 46 - Dados da resposta (POST: Consultar conta)

5.3 Decisões de implementação

Explicar nas subsecções seguintes as principais decisões tomadas antes da codificação programática do sistema.

5.3.1 Tecnologias usadas

Para a implementação deste projeto foi utilizado [Laravel Sanctum](#) que foi utilizado para as seguintes funcionalidades:

- Login/Registo dos utilizadores;
- Autenticação de API através de tokens;
- Disponibilização de cookies e a encriptação das mesmas.

A utilização do Sanctum automatizou o processo de criação e início de sessão dos utilizadores registados no sistema e ao manter a sessão do utilizador que se encontra com sessão iniciada no sistema ativa, visto que é disponibilizada um cookie que permite ao utilizador ficar autenticado enquanto navega pelo sistema, esta cookie é destruída ao o utilizador efetuar o *logout* do sistema, esta cookie é protegida de forma encriptada.

Name	Value	Domain	P..	Expires / Max-Age
vK8ABm2sEo149V9jdH4NW6...	eyJpdil6lkdvVGdDM1c1WmNxdjJ...	localhost	/	2023-02-06T20:52:15.662Z
laravel_session	eyJpdil6lksQXFzQ0wycXh5bDh1...	localhost	/	2023-02-06T20:52:15.662Z
XSRF-TOKEN	eyJpdil6lInoWTZMUjBFSG52a0RC...	localhost	/	2023-02-06T20:52:15.662Z

Figura 47 – Cookies

5.3.2 Definições de implementação

API

Validação de campos/dados

As validações dos campos/dados no sistema são feitos maioritariamente na parte registo e login no sistema onde é feita a validação dos dados que são passados pelo utilizador na view em que esses depois são verificados no controlador AuthController com o feedback do model, em que os dados irão ser verificados para saber se o utilizador tem as credenciais corretas ao iniciar sessão ou se o registo no sistema cumpre os requisitos propostos.

Segurança do sistema

A segurança do sistema começa na encriptação das cookies da sua sessão de login que consiste no browser do utilizador, esta encriptação é feita pelo Laravel Sanctum e não permite que sejam visualizadas quaisquer informações acerca das informações pessoais do utilizador com a sessão atualmente iniciada.

As palavras-passe dos utilizadores também são encriptadas de maneira que mesmo na base de dados no campo “password” não seja possível visualizar a mesma, esta encriptação é feita através de [Bcrypt](#).

Campos de palavra-passe e confirmação de palavra-passe preenchidos no registo de uma nova conta, neste caso a palavra-passe é “123”.

The image shows two vertically stacked rectangular input fields. Each field contains three dots, indicating that the text is masked. The fields are light blue with a thin border.

Figura 48 - Campos de palavra-passe

```
password
$2y$10$yUfrFWVWQNuoHkdTx.m6X.MZEK1DK4cxewJhZ41zaRJ...
```

Figura 49 - Campo da palavra-passe na base de dados

Front-End

Na página principal permite que o utilizador visualize todo o catálogo de jogos disponível no website.

O utilizador pode carregar num dos jogos para obter informação mais detalhada acerca de um determinado jogo ou navegar pelas outras funcionalidades do sistema.

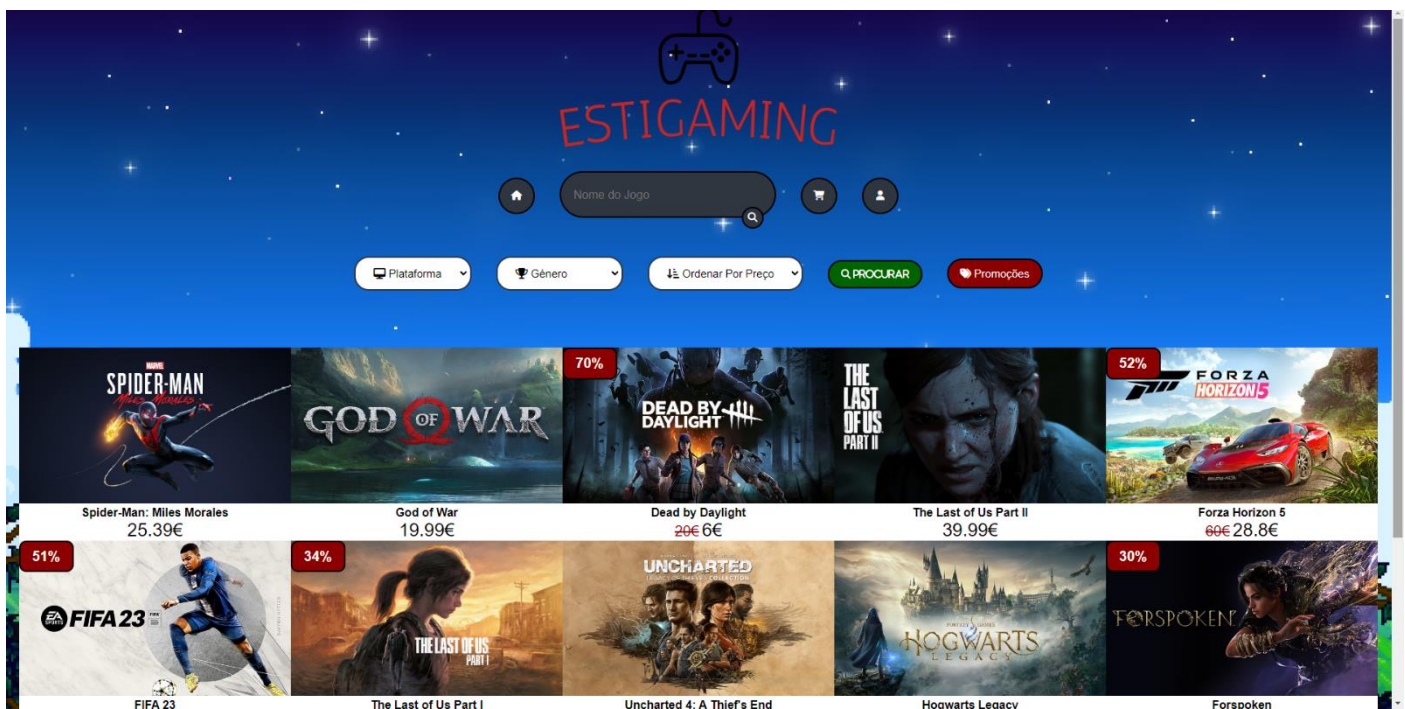


Figura 50 - Homepage

Resultados de os jogos após o utilizador inserir o nome do jogo na *input box* de pesquisa, e qualquer jogo que contenha o texto inserido no seu nome irá ser exibido.

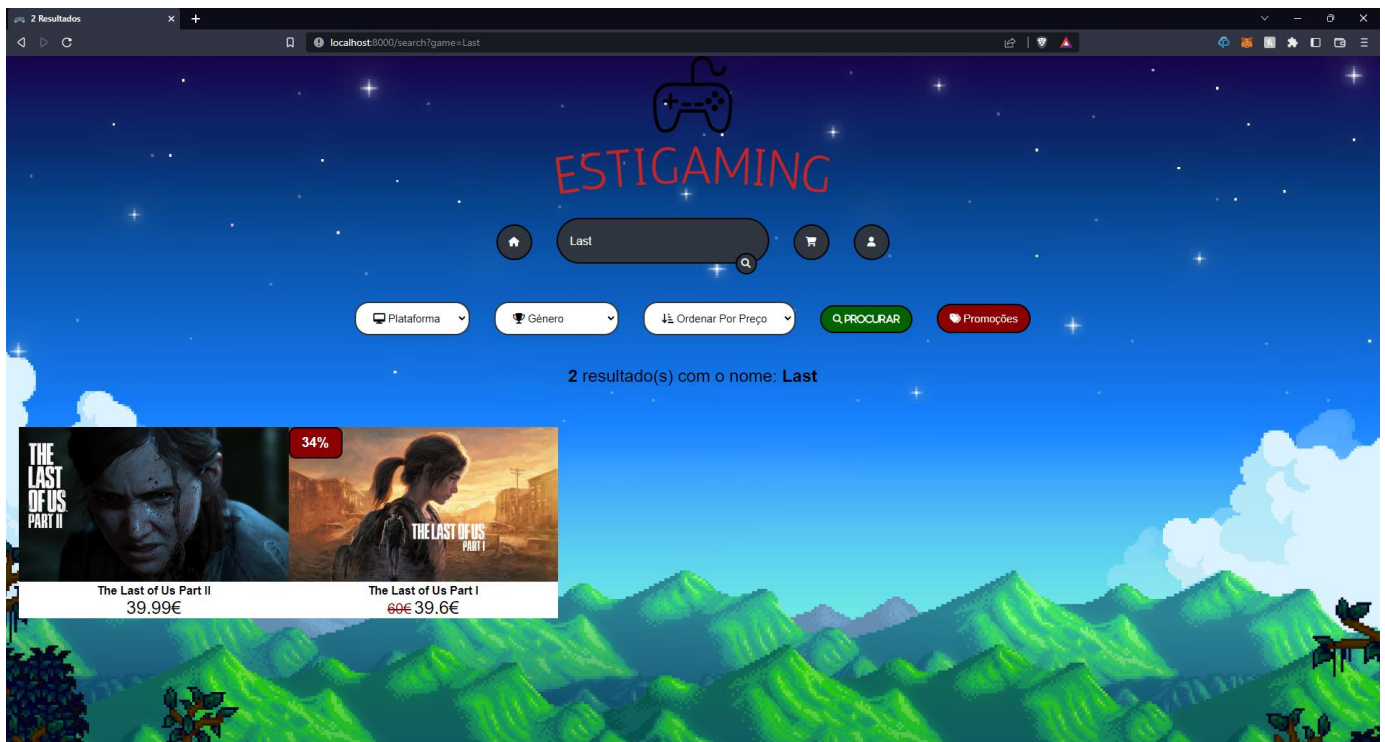


Figura 51 - Procurar jogo por nome

Resultados para a pesquisa de jogos através dos filtros disponíveis nos *dropdowns* (plataforma, género e ordenar por preço.)

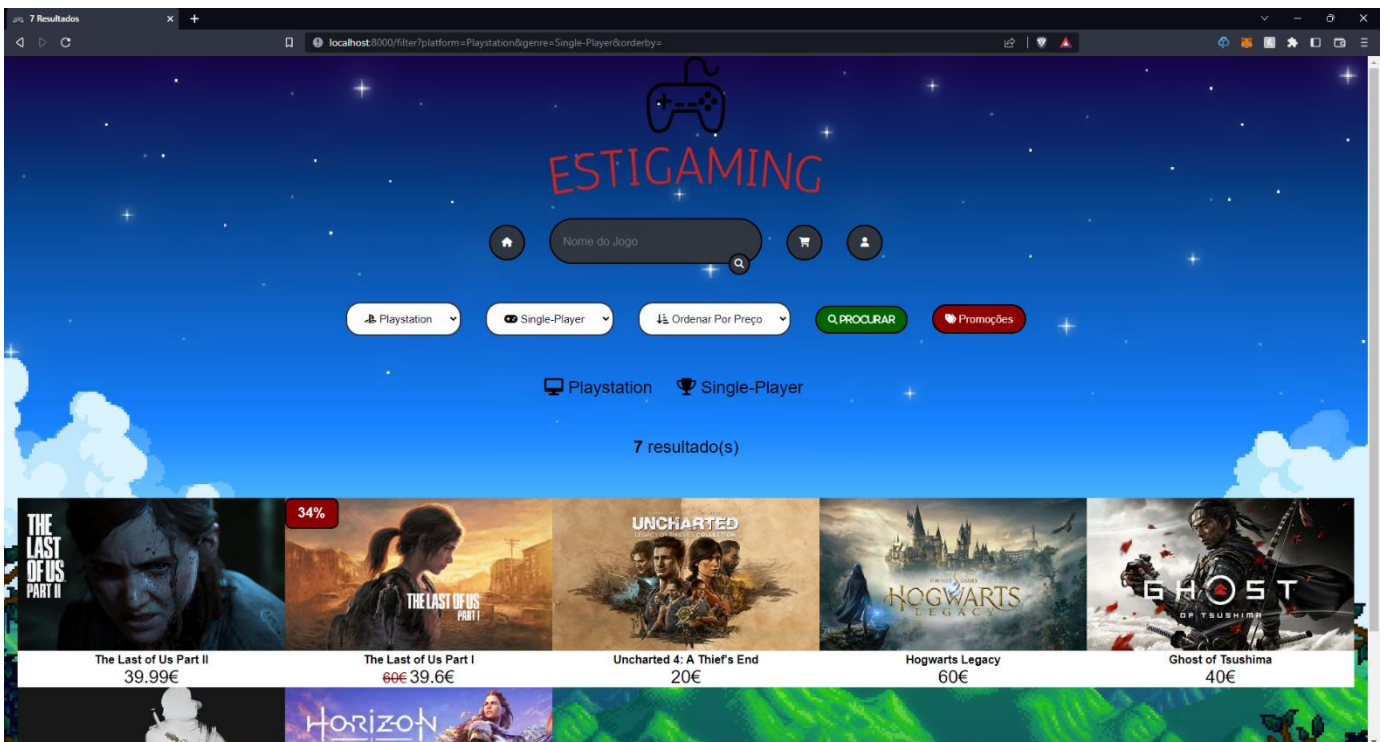


Figura 52 - Procurar por jogos através de filtros

Consultar todos os jogos que se encontram em promoção.

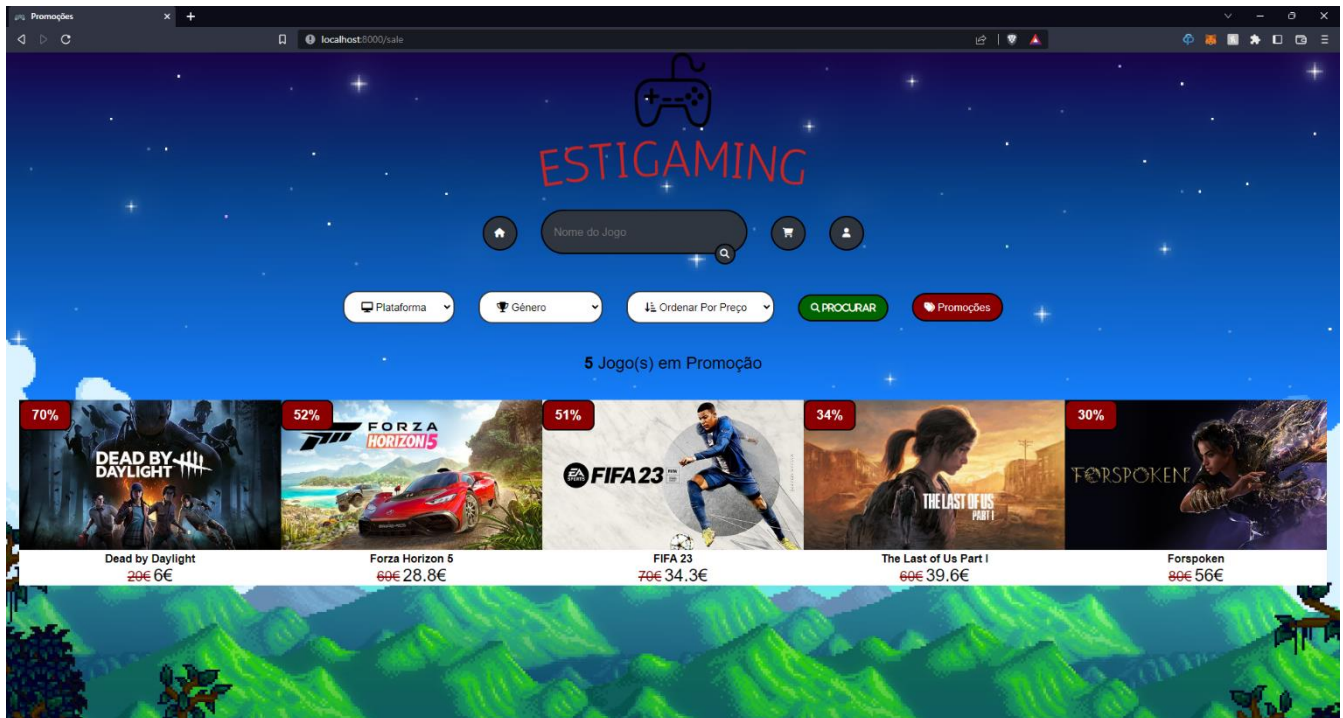


Figura 53 - Jogos em promoção

Interface após clicar num determinado jogo para obter informações mais detalhadas acerca do mesmo, ou até mesmo adicioná-lo ao carrinho de compras (caso tenha sessão iniciada).

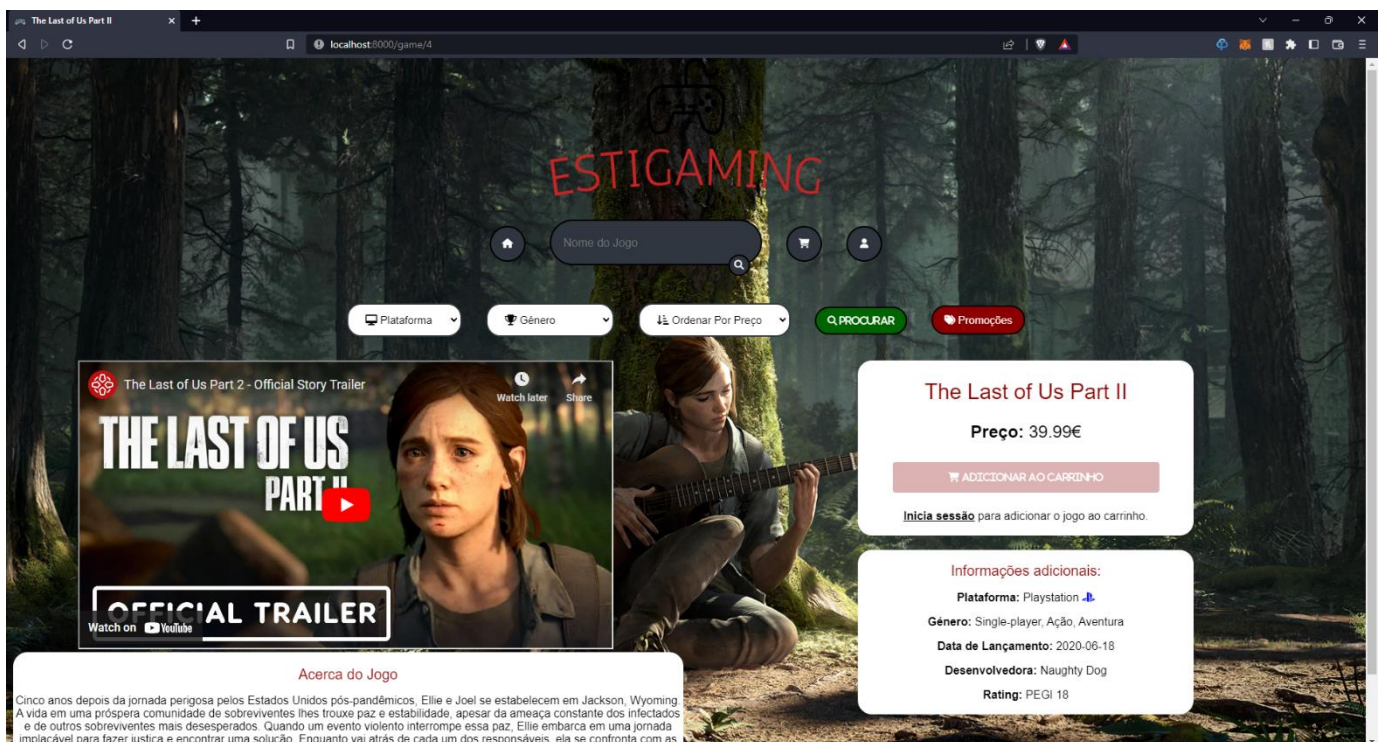


Figura 54 - Consultar detalhes de um jogo em específico

Interface para o utilizador iniciar sessão no sistema com as suas credenciais ou fazer o registo no sistema caso seja um novo utilizador.

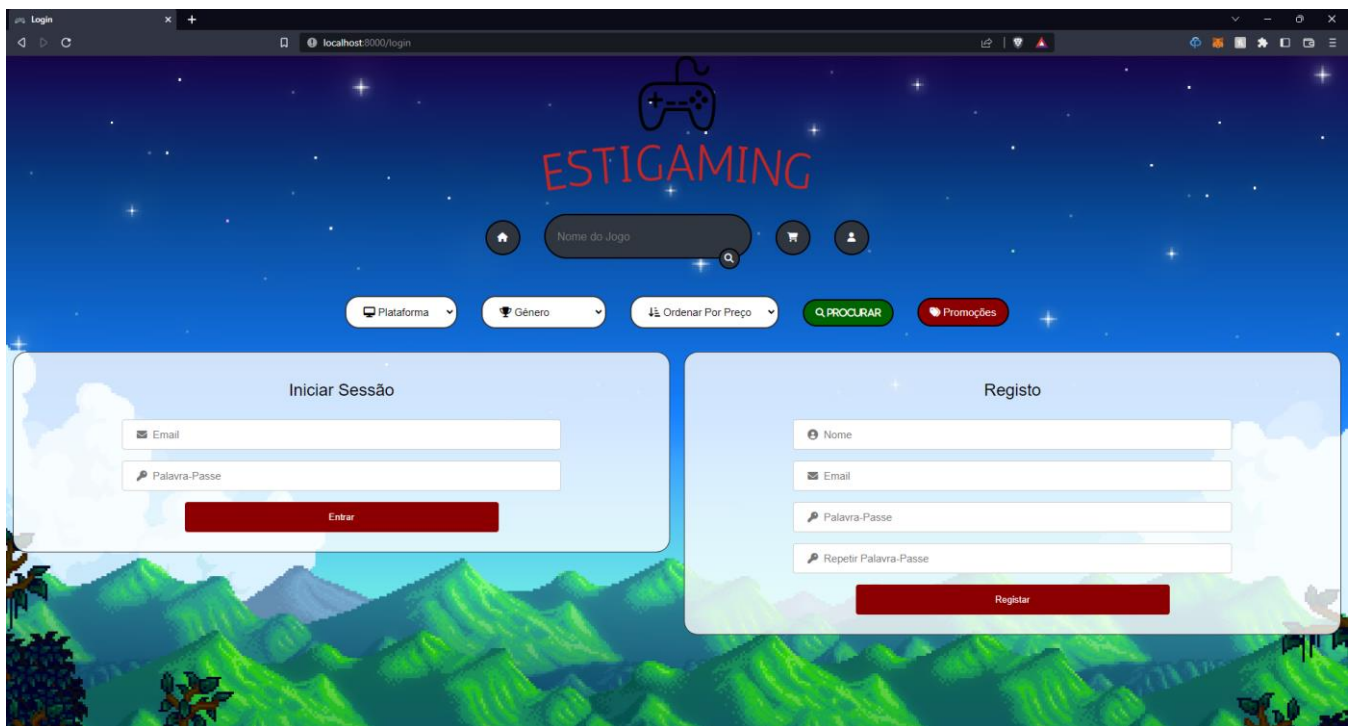


Figura 55 - Login/Registo

Jogos que se encontram no carrinho do utilizador que tem atualmente sessão iniciada no sistema (Fábio).

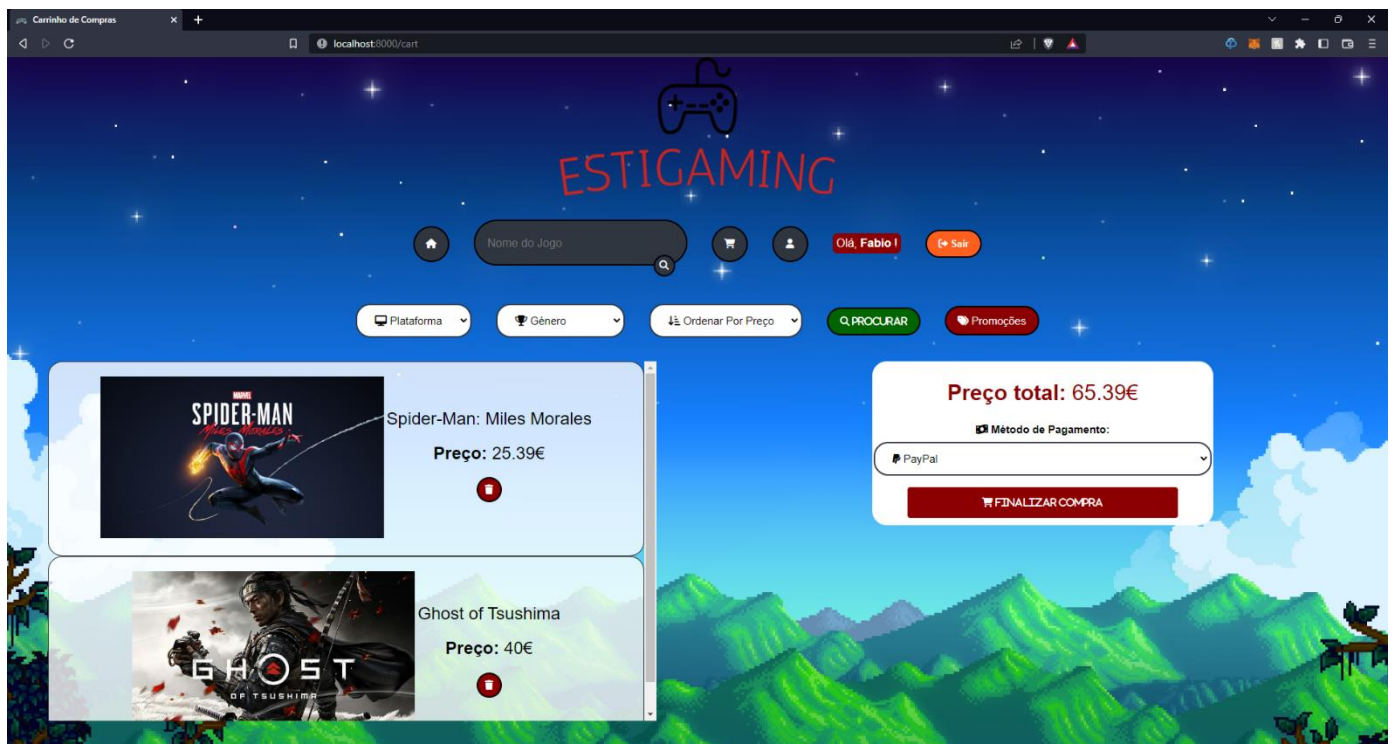


Figura 56 - Carrinho de compras

Conta do utilizador atualmente com sessão iniciada, em que é possível consultar o seu histórico de jogos comprados, assim como informação relativa às compras dos mesmos.

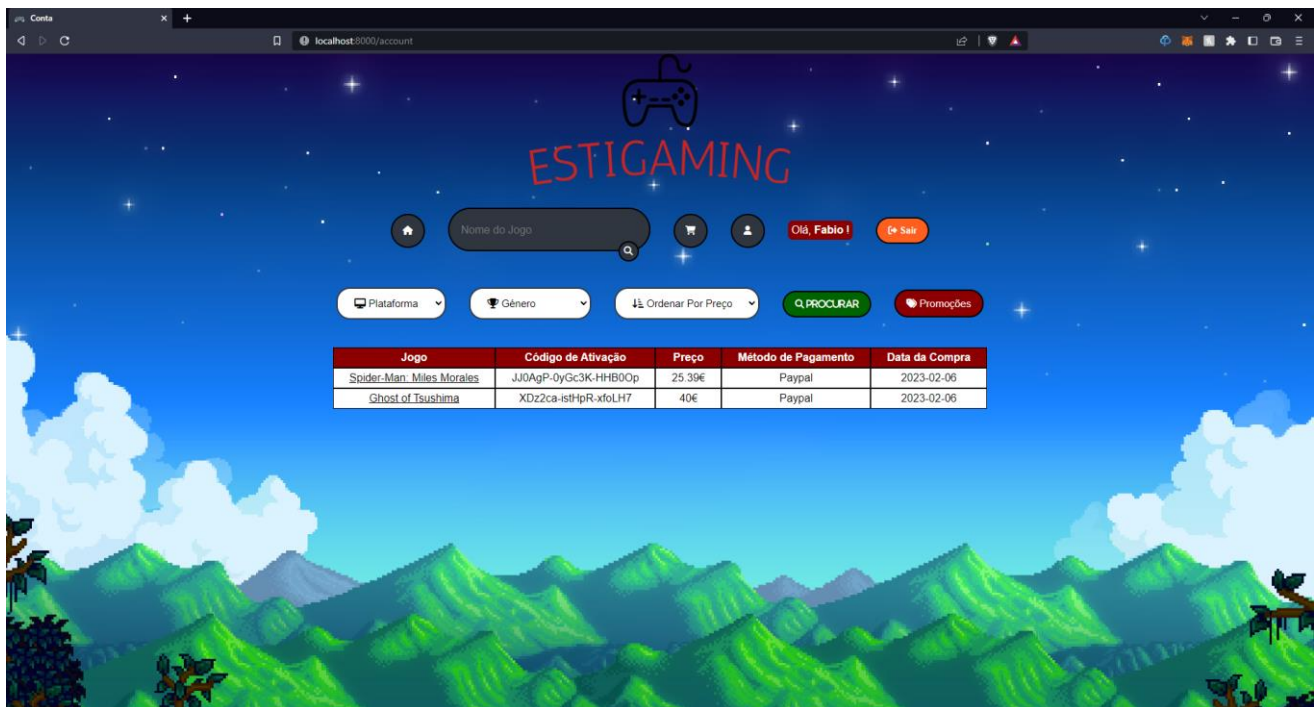


Figura 57 - Conta do utilizador

5.4 Codificação

API

Explicação dos principais e mais complexos casos de codificação, com pequenos trechos de código.

Neste caso será demonstrado o funcionamento de uma parte do código do controlador AuthController que é onde é tratada a informação acerca do login e registo visto que essa é a parte que requiere mais atenção devido às validações.

Na parte do login do sistema, o utilizador ao inserir as suas credenciais nos determinados campos de email e palavra-passe, o controller irá receber esses dados em *request* que irá ser feito um *attempt* para validar os dados em que se a verificação for bem sucedida (email e palavra-passe corresponderem a campos da BD), irá ser criada uma *token* pessoal para esse utilizador (um *token* é gerada cada vez que é efetuado o login), por fim o utilizador será redirecionado à interface em que poderá ver os detalhes sobre a sua conta, neste caso o histórico de jogos comprados pelo mesmo, caso o login esteja incorreto o utilizador é redirecionado de volta à página de login para efetuar o início de sessão novamente ou até mesmo um novo registo.

```

public function signin(Request $request)
{
    if(Auth::attempt(['email' => $request->email, 'password' => $request->password])){
        $authUser = Auth::user();
        $success['token'] = $authUser->createToken( name: 'MyAuthApp')->plainTextToken;
        $success['name'] = $authUser->name;

        return redirect()->route( route: 'account');
    }
    else{
        return view( view: 'login', ['erroLogin' => "Email e/ou Palavra-Passe incorretos."]);
    }
}

```

Figura 58 - AuthController: signin

Na parte do registo de um novo utilizador no sistema, o controlador irá fazer uma validação para verificar que todos os campos foram preenchidos e não deixados em branco, irá verificar que o email inserido é único e não corresponde ao de nenhum utilizador já existente no sistema e também que ambas as palavras-passe coincidem uma com a outra.

Caso algum dos casos acima não se verifique o utilizador será redirecionado de volta à página de login, caso contrário o utilizador consegue registar a sua conta e a palavra-passe é encriptada através do *Bcrypt* e os seus dados são inseridos na base de dados e é criada uma *token* pessoal pertencente a esse utilizador.

Por fim o utilizador é redirecionado à página de login em que irá aparecer uma mensagem que a criação do registo foi efetuada com sucesso e poderá então efetuar o início de sessão com essas mesmas credenciais.

```

public function signup(Request $request)
{
    $validator = Validator::make($request->all(), [
        'name' => 'required',
        'email' => 'required|email|unique:users,email',
        'password' => 'required',
        'confirm_password' => 'required|same:password',
    ]);

    if($validator->fails()){
        return self::backToLoginPage($request)->withErrors($validator->errors());
    }

    $input = $request->all();
    $input['password'] = bcrypt($input['password']);
    $user = User::create($input);
    $success['token'] = $user->createToken('MyAuthApp')->plainTextToken;
    $success['name'] = $user->name;

    return view( view: 'login', ['registoSuccess' => "Registo concluido com sucesso."]);
}

```

Figura 59 - AuthController: signup

Front-End

Explicação dos principais casos de codificação para os casos de uso em causa, com pequenos trechos de código.

Neste caso irá ser abordado a *view* da informação mais detalhada acerca de um determinado jogo, ou seja, quando o utilizador clica em determinado jogo (game.blade.php).

Todas as *views* seguem basicamente a mesma fórmula do *controller* pedir informação ao *model* e popular as *views* através dessa mesma informação.

Através de uma condição *if* que irá verificar se o utilizador esteja autenticado no sistema ou não, caso o utilizador esteja com sessão iniciada, o botão para adicionar ao carrinho irá estar disponível para adicionar, caso contrário irá estar inativo e pedir ao utilizador que se autentique.

Caso o utilizador interaja com o botão (estando com sessão iniciada), a informação relativa ao será enviada para o *controller* através de um *request* que será posteriormente comunicado com o *model*, será também gerada a chave de forma aleatória como podemos ver no campo “game_key” do input, para simular a chave de ativação de um videojogo em determinada plataforma.

```
@if (Auth::user())
    <form action="/game/{{ $game->id }}/addToCart" method="POST">
        <input type="hidden" id="user_id" name="user_id" value="{{ Auth::user()->id }}">
        <input type="hidden" id="game_id" name="game_id" value="{{ $game->id }}">
        <input type="hidden" id="game_key" name="game_key" value="{{ Str::random(6) . '-' . Str::random(6) . '-' . Str::random(6) }}">
        <button class="cart_btn" type="submit"><i class="fa-solid fa-cart-shopping"> Adicionar ao carrinho</i></button>
    </form>
@else
    <button disabled class="cart_btn"><i class="fa-solid fa-cart-shopping"> Adicionar ao carrinho</i></button>
    <p><a class="game_account_txt" href="/login"><b><u>Inicia sessão</u></b></a> para adicionar o jogo ao carrinho.</p>
@endif
```

Figura 60 - game.blade.php

4. Conclusão e Trabalho Futuro

Com este projeto adquiri bastantes conhecimentos no que toca ao uso da framework Laravel para o desenvolvimento de backend e os métodos utilizados para acompanhar o seu desenvolvimento.

Concluí que apesar de algumas dificuldades é sempre possível realizar um projeto desde que haja foco e determinação em querer aprender o porquê de determinados erros para perceber a lógica e poder ser implementado da forma mais correta de maneira a perceber o porquê do código funcionar dessa mesma maneira.

Posto isto, gostei bastante de utilizar a arquitetura MVC, achei que foi uma maneira mais estruturada de programar e implementar este projeto.

Por fim, foi um projeto bem executado onde se concluíram os objetivos esperados como foram descritos na parte de desenho/análise desta unidade curricular visto que tudo o que foi referido foi implementado.